2017

Efficient Reasoning Algorithms for Fragments of Horn Description Logics

David Carral
Wright State University

EFFICIENT REASONING ALGORITHMS FOR FRAGMENTS OF HORN DESCRIPTION LOGICS

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

By

DAVID CARRAL
Master of Science, Wright State University, 2012
Bachelor of Science, Universidad Pontificia de Salamanca, 2011

2017
Wright State University

WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

October 12, 2016

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY `David Carral` ENTITLED `Efficient Reasoning Algorithms for Fragments of Horn Description Logics` BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF `Doctor of Philosophy`.

Pascal Hitzler, Ph.D.
Dissertation Director

Mateen Rizki, Ph.D.
Department Chair

Committee on
Final Examination

Bernardo Cuenca Grau, Ph.D.

Pascal Hitzler, Ph.D.

Krishnaprasad Thirunarayan, Ph.D.

Michael L. Raymer, Ph.D.

Robert E. W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

ABSTRACT

Carral, David. PhD. Department of Computer Science and Engineering, Wright State University, 2017. Efficient Reasoning Algorithms for Fragments of Horn Description Logics.

We characterize two fragments of Horn Description Logics and we define two specialized reasoning algorithms that effectively solve the standard reasoning tasks over each of such fragments. We believe our work to be of general interest since (1) a rather large proportion of real-world Horn ontologies belong to some of these two fragments and (2) the implementations based on our reasoning approach significantly outperform state-of-the-art reasoners. Claims (1) and (2) are extensively proven via empirically evaluation.

Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

Amongst the many people I have had the pleasure to work with throughout these past years, there are three without whom I would not have been able to successfully finish this doctorate degree. The first of them is Dr. Bernardo Cuenca Grau, who has on many occasions attempted to teach me how to clearly structure and organize ideas, an instrumental skill I still hope to pick up some day. The second person I would like to recognize is Dr. Cristina Feier, my closest and most esteemed collaborator thus far! Many-a-deadlines I would have missed, had you not been there to help me out! And, last but not least, I would like to dearly thank my advisor Dr. Pascal Hitzler, who was unfortunate enough to have to mentor me during these most melodramatic early years of my adult life and, nevertheless, made an outstanding job out of it.

Finally, I would like to mention the other members of my dissertation committee, Dr. Krishnaprasad Thirunarayan and Dr. Michael L. Raymer. Thank you very much for seeing me through these last steps of my PhD.

On the more formal side, this work was partially supported by the National Science Foundation under award 1017225 "III: Small: TROn – Tractable Reasoning with Ontologies." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Furthermore, the author was also partially funded by the "La Caixa" foundation fellowship for postgraduate studies.

1

Introduction

On many applications, machines are primarily used to store, transmit and display information for human consumption, but often this content is "incomprehensible" to the machines themselves. Amongst other reasons, this is because the machines lack the necessary (and basic) necessary background to "understand" such content.

For example, if you ask the server hosting the Wikipedia page of King Hamlet "What is the name of the uncle of Prince Hamlet?," even if it could somehow correctly interpret the question, it still could not produce the correct answer. This is because, such an answer is not explicitly included in the webpage. Nevertheless, the webpage does contain (at the time of this write-up) the following relevant pieces of information: (a) "Prince Hamlet is the son of King Hamlet" and (b) "Claudius the brother of King Hamlet." Upon reading (a) and (b), any human would readily ascertain the answer of the previously poised question; namely, that (c) "Claudius is the uncle of Prince Hamlet." The server, lacking an "understanding" of the entities uncle, brother and father and their relation, is unable to come up with such a conclusion.

The previous issue can be addressed making use of *Description Logics* (DL); a knowledge representation family of languages which may be employed to encode information about the relationships between a set of entities within a given domain. Using DL languages, we can encode knowledge about a particular domain into *axioms* which, in turn, can be grouped into larger structures, referred to as *ontologies*. After crafting such ontologies, we can employ automated *reasoning algorithms* to derive implicit knowledge that follows from the explicitly stated assertions.

For example, we may use DL axioms (1.1) and (1.2) to convey facts (a) and (b), respectively.

$$\mathsf{HasFather}(\mathsf{princeHamlet}, \mathsf{kingHamlet}) \tag{1.1}$$

$$\mathsf{HasBrother}(\mathsf{kingHamlet}, \mathsf{claudius}) \tag{1.2}$$

Furthermore, to provide the machine with some kind of "understanding" of the relation between the entities uncle, brother and father, we may employ the following axiom.

$$\mathsf{HasFather} \circ \mathsf{HasBrother} \sqsubseteq \mathsf{HasUncle} \tag{1.3}$$

Intuitively the previous axiom indicates that, if somebody has a father who has a brother, then this brother is the uncle of that "somebody" (many more such examples will be presented in Section 2).

Axiom (4) HasUncle(princeHamlet, claudius), which can be regarded as an answer to the previously poised query, is a conclusion of the ontology containing axioms (1.1-1.3) and can be automatically by a machine by employing a *reasoner*; i.e., an implementation of the aforementioned reasoning algorithms.

1.1 Broader Impact

With the example provided in the previous section, we intend to instruct the reader about the potential of knowledge representation languages. Nevertheless, this is only a toy example which does not accurately showcase the usefulness of these technologies. We proceed thus, to elaborate more broadly about the use of these languages in different domain applications.

The use of DL ontologies has been steadily gaining in popularity and is used in countless application areas, such as social network analysis [Fan 2012], city data processing [Lécué et al. 2014] and network traffic analysis [Barrett et al. 2000]. Perphaps, the most relevant and widespread application of DL takes place within the context of the Semantic Web [Hitzler et al. 2009], an extension of the World Wide Web where structure and meaning are provided in the hopes of making the information in web pages understandable.

The Semantic Web is an extension of the Web through standards by the World Wide Web Consortium which promote common data formats and exchange protocols on the Web. One of such standards is the Web Ontology Language (OWL) which provides a computer based syntax for DL languages. Using OWL, ontologies can be encoded in a way and manner that the information such as the one presented in the previous section can be encoded in the content of webpages. Alas, the use of OWL would allow use to solve the issue presented in the previous example!

As with DLs, the use of OWL allows for the automated inference of implicit information from explicit statements. It is thus, of the utmost importance to develop efficient reasoning algorithms which are scalable over large ontologies. In this thesis we propose the use of two of such reasoning algorithms which have the potential of improving the performance of many existing real-world applications.

1.2 Problem Statement

Hopefully by now, the reader is somewhat convinced of the usefulness of knowledge representation languages and automated reasoning techniques. We proceed in this section to argue why the development of reasoning algorithms is non-trivial and challenging endeavor.

First of, note that, reasoning is over many expressive DL languages theoretically hard. In fact, for all DL languages which which cannot be characterized within the tractable fragments; namely, $\mathcal{EL}^{++}$[Baader et al. 2005a], DL-Lite$_R$ [Calvanese et al. 2007] and DLP [Grosof et al. 2003]; reasoning is at least ExpTime-hard.

On this thesis, we focus on the theoretical definition and implementation of very efficient reasoning

algorithms for customized fragments of Horn-$\mathcal{SROIQ}$, a DL language with 2-ExpTimecomplexity. Horn-$\mathcal{SROIQ}$ may be intuitively understood as the DL language which does not allow for the use of non-deterministic logic constructors such as disjunction or certain forms of negation (this language will be intuitively introduced and formally defined in Sections 2 and 3, respectively). Furthermore, our reasoning algorithms may not only be employed to solve standard reasoning tasks, such as satisfiability or classification, but may also the more complicated problem of conjunctive query answering.

Moreover, we would like to remark that the problem tackled in this thesis is not simply theoretically hard. Despite the fact that many algorithms have been formally defined in the past to solve conjunctive query answering over DL ontologies [Stefanoni et al. 2014; Rudolph and Glimm 2014; Ortiz et al. 2011], we are not aware of any practical implementations of such procedures for expressive fragments DL. As part of the output of this thesis, we implement two different reasoning algorithms that are quite efficient and outperform existing state-of-the-art implementations.

As briefly mentioned in the previous paragraph, our reasoning algorithms may not be applied to all Horn-$\mathcal{SROIQ}$ ontologies. Therefore, we empirically verify that many ontologies can be defined within our newly defined fragments, which in turn validates the usefulness of our approach (more results to that effect may be found in Sections 4 and 5).

1.3 Structure

We proceed to briefly describe the structure of this thesis which includes five more further sections.

- Section 2: In this section we informally introduce DL languages in an attempt to convey an intuitive understanding of this standards aimed at the unexperienced reader.
- Section 3: In this section we formally define DL languages and introduce several notions used in the subsequent sections of the thesis.
- Sections 4: This section formally presents the RCA fragment of Horn-$\mathcal{SROIQ}$ and an algorithm to reason over RCA ontologies. Furthermore, this section includes an extensive evaluation of the RCA fragment plus a comparison of the implementation of the reasoning algorithm agains other reasoners.
- Section 5: This section is similar in quite similar to the previous one in terms of content, but it considers the RSA fragment of Horn-$\mathcal{SROIQ}$ instead of RCA.
- Section 6 This section lists the conclusions of this thesis and elaborates about possible further work.

2

An Intuitive Introduction to Description Logics

Description logics (DL) is a family of knowledge representation formalisms widely used in ontological modeling. All DL languages are fragments of first-order logic and as such, they are equipped with a formal semantics. Thus, the use of DL enables a precise specification of the understanding of DL logical formulas (also referred as axioms) which, apart from allowing the exchange of knowledge without, ambiguity also enable formal deduction. Given a set of DL formulas (also referred to as an ontology), we can employ logical deduction to infer additional information from the facts explicitly included in such ontology. The derivation of implicitly entailed formulas from a given set of explicitly stated axioms is often referred to as reasoning, which is a challenging and relevant task employed in many real-world applications.

The main contribution of this thesis is the definition of two fragments of Horn DL (i.e., the subset of DL which does not allow for the use of non-deterministic logical constructors such as disjunction) and the presentation of two corresponding algorithms to solve reasoning tasks over such fragments. As our main research hypothesis, we aim to empirically verify that our algorithms significantly outperform state-of-the-art systems and that our newly defined fragments characterize a large proportion of real-world Horn DL ontologies.

The remainder of the thesis is structured as follows. In the rest of this chapter we present an informal introduction to DL. In the subsequent chapter we present a formal definition of DL which also includes the definition of many preliminary notions that will be used in further sections. Chapters 4 and 5 describe the two aforementioned fragments of Horn DL and the algorithms to reason over each of such fragments, respectively. On the last chapter of this thesis we list our conclusions and elaborate about possible further work.

2.1 An Informal Introduction to DL

Along this section, we provide an informal and intuitive introduction to DL. See the following section for a formal and precise definition of the syntax and semantics of this family of knowledge represen-

tation languages.

2.1.1 Entities

DL languages provide means to model the relations between *entities* in a *domain of interest*. In DL there are three kinds of entities: *concepts*, *roles* and *individual names*. Concepts represent sets of individuals, roles represent binary relations between individuals and individual names represent single individuals in the domain. Readers familiar with first-order logic will recognize these entitites as unary predicates, binary predicates and constants.

For example, if we decide to model the domain of their family relationships, we may use concepts such as Female to represent the set of all female individuals, respectively. Furthermore, we may use roles such as IsParentOf to represent the (binary) relationship between parents and their children. Finally, we may use individual names such as joe to represent the domain individual Joe (note the distinction between "individual names;" i.e., DL entities which represent objects in the domain of interest; and "domain individuals;" i.e., the objects in the domain of interest which these DL entities represent).

2.1.2 Logical Constructors

Combining the use of entities with the available DL logical constructors we can create more involved (and more useful) sets of individuals, which are referred as *concept expressions*. For example, we can employ the *intersection* (also called *conjunction*) *constructor* to define a concept expressions such as the following.

$$\mathsf{Female} \sqcap \mathsf{Parent} \tag{2.1}$$

The previous concept expression subsumes all the domain individuals that are in the intersection of the concepts Female and Parent; i.e., concept expression (2.1) includes a named individual if and only if it belongs to both the Female and Parent concepts. Intuitively, concept expression (2.1) coincides with our understanding of the idea motherhood.

Another way of constructing concept expressions is the use of *role restrictions*, which link concepts and roles together. For example, making use of role restrictions we can declare concept expressions such as the following.

$$\exists \mathsf{hasChild}.\mathsf{Person} \tag{2.2}$$

The previous concept expression, referred to as an *existential restriction*, characterizes every domain individual related to another domain individual in the class Person via the property hasChild. Intuitively, this expression coincides with the idea of parenthood.

To represent the set of individuals all of whose children are female, we may employ *universal restrictions* such as the following.

$$\forall \mathsf{IsParentOf}.\mathsf{Female} \tag{2.3}$$

The previous concept expression characterizes every individual in domain such that, if this individual is connected to some other individual via role IsParentOf, then the latter is subsumed by the class Female. Note that, such a concept also subsumes those individuals that have no children at all. If this meaning is not intended, one can describe the individuals with some children and with all their children being female with the following concept.

$$\exists \mathsf{IsParentOf}.\top \sqcap \forall \mathsf{IsParentOf}.\mathsf{Female} \tag{2.4}$$

The first operand in the conjunction indicates that this concept expression may only subsume individuals with at least some children. Furthermore, the second operand indicates that concept expression only subsumes individuals who only have daughters.

Functional restrictions allow us to restrict the number of individuals that be reached via a given role.

$$\leq 1\, \mathsf{IsParentOf}.\top \tag{2.5}$$

The previous concept expression, referred to as an *at-least* restriction, is employed to characterize domain individuals with at most one children.

Finally, *nominals* may be employed to denote classes containing a single individual.

$$\{\mathsf{joe}\} \tag{2.6}$$

The previous class expression denotes the set only containing the domain individual Joe (represented by the name individual joe).

Apart from boolean constructors, role restrictions and nominals, which are used to build concept expressions, we can also make use *role constructors* to construct *role expressions*. For example, we can make use of the *role inverse constructor* to define a role expression such as the following.

$$\mathsf{IsParentOf}^{-} \tag{2.7}$$

The previous role expression comprehends the inverse of the relation subsumed by the role IsParentOf; i.e., if some individual is connected to another via IsParentOf, then the latter is connected to the former via the role expression $\mathsf{IsParentOf}^{-}$.

Finally, we may use *role chains* to define relations that result from the composition of two or more existing roles.

$$\mathsf{hasParent} \circ \mathsf{hasBrother} \tag{2.8}$$

The previous role expression subsumes pairs of individuals for which the first individual in the pair is connected to a third individual via role hasParent which, in turn, is connected to the second individual via role hasBrother.

2.1.3 Axioms

Concept and role expressions may be employed in the construction of *axioms*, which are formulas that capture partial knowledge about the domain of interest. As customary, we separate DL axioms into two groups: *assertional* or *ABox axioms*, and *terminological* or *TBox axioms*.

2.1.3.1 ABox axioms

Some ABox axioms, such as the following, capture knowledge the concepts to which named individuals belong to.

$$\mathsf{Mother}(\mathsf{mary}) \tag{2.9}$$

$$\mathsf{Male}(\mathsf{joe}) \tag{2.10}$$

The previous axioms indicate that Mary –the domain individual represented by the named individual mary– is a mother and joe –the domain individual represented by the named individual joe– is male, respectively. ABox axioms such as (2.9) and (2.10) are referred to as *concept assertions*.

ABox axioms may also be used to describe relations between named individuals.

$$\mathsf{IsParentOf}(\mathsf{mary}, \mathsf{joe}) \tag{2.11}$$

$$\mathsf{SisterOf}(\mathsf{mary}, \mathsf{mike}) \tag{2.12}$$

The previous ABox axioms are referred to *role assertions*.

Although common sense may indicate that Mary and Joe are different individuals, this fact does not logically follow from the previous axioms. Note that, DLs do not make the *unique name assumption* (UNA), and so different named individuals may refer to the same domain individual.

2.1.3.2 TBox axioms

TBox (terminological box) axioms allow us to define relations at a terminological level; i.e., relationships between concept and role expressions. For example, we may use TBox axioms such as the following to indicate that Mother is a subclass of Parent.

$$\mathsf{Mother} \sqsubseteq \mathsf{Parent} \tag{2.13}$$

Axioms such as (2.13) are often referred to as *subclass axioms*.

In a similar manner, we may also use TBox axioms to enforce subclass and equivalence relations over role expressions.

$$\mathsf{hasParent} \circ \mathsf{hasBrother} \sqsubseteq \mathsf{hasUncle} \tag{2.14}$$

$$\mathsf{IsMotherOf} \sqsubseteq \mathsf{IsParentOf} \tag{2.15}$$

Axioms (2.14) and (2.15) intuitively indicate the following: the composition relation resulting from concatenating the roles hasParent and hasBrother is a subrelation of hasUncle and the role IsMotherOf is a subrelation of the rule IsParentOf.

2.1.4 Ontologies

An ontology is a set of axioms which often satisfies some additional syntactic restrictions. Such restrictions are dependent on the particular DL language employed and will be described at length in the following section.

2.1.5 Reasoning Tasks

Given some ontology, the task of computing the axioms and/or answers to queries entailed by such ontology is referred to as *reasoning*. There exist several different types of reasoning tasks which are grouped as follows.

1. The *standard reasoning tasks*: Amongst others, these include *ontology satisfiability*, i.e., checking whether an ontology entails a logical inconsistency; *classification*, i.e., computing all axioms of the form $A \sqsubseteq B$, where A and B are concepts, entailed by an ontology; and *ABox retrieval*, i.e., computing all ABox assertions entailed by an ontology.

2. *Conjunctive query answering*: This is the task of determining whether a sequence of individuals is an answer of a conjunctive query with respect to a given ontology. A CQ is a formula such as $\exists x, z : \mathsf{attendsCourse}(x, y) \wedge \mathsf{Course}(y) \wedge \mathsf{taughtBy}(y, z) \wedge \mathsf{supervisedBy}(\mathsf{x}, \mathsf{z})$ (intuitively, the answer of the previous query would include all pairs of students and supervisors such that the student attends a course taught by the supervisor).

3

Formal Preliminaries

3.1 Existential Rules

Even though reasoning over programs with existential rules is not the main topic of this thesis, some of the reasoning algorithms presented in later sections are based on this paradigm and thus, we formally introduce it in this chapter.

Let $\mathbf{C}$, $\mathbf{F}$, $\mathbf{V}$ and $\mathbf{P}$ be pairwise disjoint and infinite countable sets of *constants*, *function symbols*, *variables* and *predicates*, respectively, where every function symbol and predicate s is associated with some *arity* $ar(s) \geq 1$. Furthermore, $\top, \bot, \approx \in \mathbf{P}$ with $ar(\top) = ar(\bot) = 1$ and $ar(\approx) = 2$. As customary, we often refer to the special predicate $\approx$ as the *equality predicate*.

The set of *terms* $\mathbf{T} \supseteq \mathbf{C} \cup \mathbf{V}$ is the minimal set such that, for every function symbol f with $ar(f) = n$ and every sequence of terms $t_1, \ldots, t_n$, $f(t_1, \ldots, t_n) \in \mathbf{T}$. The *depth* $dep(t)$ of a term t is defined as 0 if $t \in \mathbf{C} \cup \mathbf{V}$, and $dep(t) = t + max_{i=1}^{n} dep(t_i)$ if t is of the form $f(t_1, \ldots, t_n)$. We often abbreviate a sequence of terms $t_1, \ldots, t_n$ as $\vec{t}$ and identify such a sequence with the set $\{\vec{t}\}$. A term t' is a *subterm* of another term t if and only if $t' = t$, or $t = f(\vec{t})$ and t' is a subterm of some $s \in \vec{t}$; if additionally $t' \neq t$, then t' is a *proper subterm* of t. A term t is *n-cyclic* if and only if there exists a sequence of terms of the form $f(\vec{s_1}), \ldots, f(\vec{s_{n+1}})$ such that $f(\vec{s_{n+1}})$ is a subterm of t and, for every $i = 1, \ldots, n$, $f(\vec{s_i})$ is a proper subterm of $f(\vec{s_{i+1}})$. We simply refer to 1-cyclic terms as *cyclic*.

An *atom* is a formula of the form $p(\vec{t})$ where p is some predicate with $ar(p) = |\vec{t}|$. A *fact* is a ground atom; i.e., an atom without occurrences of variables.

A *substitution* is a function mapping terms to terms. Given a substitution σ and an atom $\alpha = p(t_1, \ldots, t_n)$, the *application of* σ *on* α is the atom $\alpha\sigma = p(\sigma(t_1), \ldots, \sigma(t_n))$. The result of the application of a substitution on a sequence of variables, a conjunction of atoms or a rule is defined in the obvious manner. Given a sequence of terms $\vec{t}$, we define $\sigma_{\vec{t}} \subseteq \sigma$ as the maximal substitution only defined for the terms in $\vec{t}$. As customary, we represent the substitution mapping t_i to u_i for all $i = 1, \ldots, n$ with the expression $[t_1/u_1, \ldots, t_n/u_n]$.

Let t be some ground term and c some constant. Let t_c be the term obtained from t by replacing every occurrence of a constant by c, i.e., $f(d, g(e))_c = f(c, g(c))$. The notation is analogously extended to facts and sets of facts.

With $\phi(\vec{x})$ we stress that $\vec{x} = x_1, \ldots, x_n$ are the free variables occurring in the formula ϕ. Furthermore, we often identify a conjunction of atoms $\phi_1 \wedge \ldots \wedge \phi_n$ with the set $\{\phi_1, \ldots, \phi_n\}$.

A *rule* is a first-order logic (FOL) formula of one of the following forms.

$$\forall \vec{x} \forall \vec{z} (\beta(\vec{x}, \vec{z}) \rightarrow \exists \vec{y} \eta(\vec{x}, \vec{y})) \tag{3.1}$$

$$\forall \vec{x} (\beta(\vec{x}) \rightarrow t \approx u) \tag{3.2}$$

In the above, β and η are non-empty conjunctions of atoms which do not contain occurrences of function symbols or equality; $\vec{x}$, $\vec{y}$ and $\vec{z}$ are pairwise disjoint; $\vec{x}$ is non-empty; and $t, u \in \vec{x} \cup \mathbf{C}$. To simplify the notation, we frequently omit the universal quantifiers from rules. As customary, we refer to rules of the forms (3.1) and (3.2) as *tuple generating dependencies* (*TGDs*) and *equality generating dependencies* (*EGDs*), respectively.

Given a set of rules $\mathcal{R}$, we define $\mathcal{R}^{\exists}$ and $\mathcal{R}^{\forall}$ as the sets of all the TGDs in $\mathcal{R}$ which do and do not contain existentially quantified variables, respectively. Moreover, let $\mathcal{R}^{\approx}$ be the set of all EGDs in $\mathcal{R}$. A *program* is a tuple $\langle \mathcal{R}, \mathcal{I} \rangle$ where $\mathcal{R}$ is a set of rules and $\mathcal{I}$ is an *instance*; i.e., a finite set of equality and function free facts. Given a program $\langle \mathcal{R}, \mathcal{I} \rangle$, we assume that, without loss of generality, every predicate in $\mathcal{R}$ occurs in $\mathcal{I}$ and every constant in $\mathcal{R}$ occurs in $\mathcal{I}$. Often, we abuse notation and identify the set of rules and facts $\mathcal{R} \cup \mathcal{I}$ with the program $\langle \mathcal{R}, \mathcal{I} \rangle$.

For the remainder of the paper, we assume that $\top$ and $\bot$ are treated as ordinary unary predicates when dealing with programs and that the semantics of $\top$ is captured explicitly in any program $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$ by including the rule $p(x_1, \ldots, x_n) \rightarrow \top(x_1) \wedge \ldots \wedge \top(x_n)$ in $\mathcal{R}$ for every predicate p with arity n occurring in $\mathcal{P}$.

We will later employ skolemization to define the consequences of a TGD over a set of facts. The *skolemization* $sk(\rho)$ of some TGD $\rho = \beta(\vec{x}, \vec{z}) \rightarrow \exists \vec{y} \eta(\vec{x}, \vec{y})$ is the rule $\beta(\vec{x}, \vec{z}) \rightarrow \eta(\vec{x}, \vec{y}) \sigma_{sk}$ where σ_{sk} is a substitution mapping every $y \in \vec{y}$ into $f^y_\rho(\vec{x})$ where f^y_ρ is a fresh function unique for every variable y and TGD ρ.

3.2 Horn Description Logics

In this section, we formally define the syntax and semantics of several Horn DL languages considered across this paper. More precisely, we define Horn-$\mathcal{SROIQ}$ and Horn-$\mathcal{SHOIQ}$ [Krötzsch et al. 2013; Ortiz et al. 2010], $\mathcal{EL}^{++}$[Baader et al. 2005b], DL-Lite$_R$ [Calvanese et al. 2007] and DLP [Grosof et al. 2003]. Readers familiar with the Web Ontology Language (OWL) [Hitzler et al. 2009] may recognize these languages as the logics that underpin the DL, EL, QL and RL standards, respectively. We assume basic familiarity with the topic and otherwise refer to the literature for further details: for a thorough theoretical introduction see [Baader et al. 2007]; an extended introduction to DL and Semantic Web technologies is provided in [Hitzler et al. 2009], where also the relationships between DL and the different OWL standards are explained in detail. Without loss of generality, we restrict our attention to ontologies in a normal form close to those in [Krötzsch et al. 2007; Ortiz et al. 2010].

A *DL signature* is a tuple $\langle N_C, N_R, N_I \rangle$ where N_C, N_R and N_I are countable and mutually disjoint

$$\prod_{i=1}^{n} A_i \sqsubseteq B \longrightarrow \bigcap_{i=1}^{n} A_i^{\mathcal{I}} \subseteq B^{\mathcal{I}} \tag{3.3}$$

$$A \sqsubseteq \exists R.B \longrightarrow x \in A^{\mathcal{I}} \rightarrow \exists y((x,y) \in R^{\mathcal{I}} \wedge y \in B^{\mathcal{I}}) \tag{3.4}$$

$$\exists R.A \sqsubseteq B \longrightarrow (x,y) \in R^{\mathcal{I}} \wedge y \in A^{\mathcal{I}} \rightarrow x \in B^{\mathcal{I}} \tag{3.5}$$

$$\top \sqsubseteq \forall R.B \longrightarrow (x,y) \in R^{\mathcal{I}} \rightarrow y \in B^{\mathcal{I}} \tag{3.6}$$

$$A \sqsubseteq \leq 1\, R.B \longrightarrow x \in A^{\mathcal{I}} \wedge (x,y) \in R^{\mathcal{I}} \wedge y \in B^{\mathcal{I}} \wedge (x,z) \in R^{\mathcal{I}} \wedge z \in B^{\mathcal{I}} \rightarrow y = z \tag{3.7}$$

$$A \sqsubseteq \{a\} \longrightarrow x \in A^{\mathcal{I}} \rightarrow x = a^{\mathcal{I}} \tag{3.8}$$

$$A \sqsubseteq \exists R.\{a\} \longrightarrow x \in A^{\mathcal{I}} \rightarrow (x, a^{\mathcal{I}}) \in R^{\mathcal{I}} \tag{3.9}$$

$$R \sqsubseteq S \longrightarrow R^{\mathcal{I}} \subseteq S^{\mathcal{I}} \tag{3.10}$$

$$R^{-} \sqsubseteq S \longrightarrow (x,y) \in R^{\mathcal{I}} \rightarrow (y,x) \in S^{\mathcal{I}} \tag{3.11}$$

$$R \circ S \sqsubseteq V \longrightarrow (x,y) \in R^{\mathcal{I}} \wedge (y,z) \in S^{\mathcal{I}} \rightarrow (x,z) \in S^{\mathcal{I}} \tag{3.12}$$

$$A(a) \longrightarrow a^{\mathcal{I}} \in A^{\mathcal{I}} \tag{3.13}$$

$$R(a,b) \longrightarrow (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \tag{3.14}$$

Figure 3.1: Syntax and Semantics of Axioms. In the above, $A_{(i)}, B \in N_C$, $R, S, V \in N_R$ and $a, b \in N_I$.

sets of *concepts*, *roles* and *individual names*, respectively, where we additionally assume that $\top, \bot \in N_C$. As customary, we refer to $\top$ and $\bot$ as the top and bottom concepts, respectively. For the rest of this paper we assume that a DL signature has been fixed and so omit further references to it when possible without introducing ambiguity.

An *axiom* is a formula of one of the forms given in the left hand side of Figure 3.1. Some of the axioms such figure may appear redundant, as they can be normalized away using other axioms in such figure; nevertheless, these extra axioms are useful for defining the different types of DL languages considered in this document (see Definition 3.2.1).

Axioms of the form (3.3-3.12) are referred to *terminological* or *TBox axioms*. Axioms of the form (3.13) and (3.14) are referred to *assertional* or *ABox axioms* (or simply *assertions*). A *TBox* (resp. an *ABox*) is a set of TBox (resp. ABox) axioms.

Given some TBox $\mathcal{T}$, we define the relation $\sqsubseteq^*_{\mathcal{T}}$ over $N_R \cup \{R^- \mid R \in N_R\}$ as the minimal transitive and reflexive relation such that $R \sqsubseteq^*_{\mathcal{T}} S$ and $R^- \sqsubseteq^*_{\mathcal{T}} S^-$ if $R \sqsubseteq S \in \mathcal{T}$, and $R^- \sqsubseteq^*_{\mathcal{T}} S$ and $R \sqsubseteq^*_{\mathcal{T}} S^-$ if $R^- \sqsubseteq S \in \mathcal{T}$, where $R, S \in N_R$.

Definition 3.2.1 *We proceed with the definition of all the different types of TBoxes considered across this document.*

1. *A Horn-$\mathcal{SROIQ}$ TBox is a TBox.*
2. *A Horn-$\mathcal{SHOIQ}$ TBox is a TBox such that, for every axiom of the form $R \circ S \sqsubseteq V \in \mathcal{T}$, $R = S = V$.*
3. *A Horn-$\mathcal{SRIQ}$ TBox is a TBox which does not contain axioms of the form (3.8) or (3.9).*

4. *An $\mathcal{EL}^{++}$ TBox $\mathcal{T}$ is a TBox which does not contain axioms of the form (3.7) or (3.11). Furthermore, if $\mathcal{T}$ contains axioms the form $R \circ S \sqsubseteq V$ and $\top \sqsubseteq \forall V'.B$ in $\mathcal{T}$ with $V \sqsubseteq^*_{\mathcal{T}} V'$, then $\top \sqsubseteq \forall S.B \in \mathcal{T}$.*

5. *A DL-Lite$_R$ TBox $\mathcal{T}$ is a TBox which does not contain axioms of the form (3.5), (3.7), (3.8), (3.9) or (3.12). Furthermore, for every axiom in $\mathcal{T}$ of the form (3.3), $n = 1$.*

6. *An DLP TBox is a TBox which does not contain axioms of the form (3.4).*

An $\mathcal{L}$ ontology, where $\mathcal{L}$ is one of the previously discussed DL languages, is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is an $\mathcal{L}$ TBox.

As with programs, we often abuse notation and simply identify the set of axioms $\mathcal{T} \cup \mathcal{A}$ with the ontology $\langle \mathcal{T}, \mathcal{A} \rangle$.

Note that, we disregard certain syntactic restrictions, such as simplicity of roles occurring with numeric quantifiers or role regularity, considered in other definitions of DL [Horrocks et al. 2006]. Such restrictions, imposed in order to maintain decidability of tableau based reasoning algorithms, are unnecessary for our specialized algorithms and thus, we ignore them.

Without loss of generality, instead of dealing with conjunctive query (CQ) answering, we restrict our attention to the simpler task of CQ entailment of *boolean conjunctive queries* (BCQs). This is without loss of generality since CQ answering can be reduced to checking entailment of BCQs. A *BCQ*, or simply a *query*, is a formula of the form $\exists \vec{y} \eta(\vec{y})$ where η is a conjunction of atoms not containing occurrences of constants, function symbols nor $\approx$.

As customary, the semantics of ontologies is given through the definition of *interpretations*. An interpretation $\mathcal{I}$ of some ontology $\mathcal{O}$ is a tuple $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set, referred to as the *domain of* $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is a function that maps each individual, concept and role in $\mathcal{O}$ to an element, a subset and a binary relation in $\Delta^{\mathcal{I}}$, respectively, such that $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$.

Let $\mathcal{I}$ be an interpretation of some ontology $\mathcal{O}$. We say that $\mathcal{I}$ *satisfies an axiom* α, written $\mathcal{I} \models \alpha$, if the corresponding condition shown in Figure 3.1 holds. Furthermore, we say that $\mathcal{I}$ *satisfies the ontology* $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, if $\mathcal{I} \models \alpha$ for every $\alpha \in \mathcal{O}$. If this is indeed the case, we say that $\mathcal{I}$ *is a model of* $\mathcal{O}$.

An ontology $\mathcal{O}$ is *satisfiable* if there exists at least some interpretation $\mathcal{I}$ which is also a model of $\mathcal{O}$. An ontology $\mathcal{O}$ entails an axiom α, written $\mathcal{O} \models \alpha$, if, for every model $\mathcal{I}$ of $\mathcal{O}$, $\mathcal{I} \models \alpha$.

Finally, we introduce several *reasoning tasks* which will be considered across this document. Let $\mathcal{O}$ be some ontology.

- Ontology satisfiability: Determine whether $\mathcal{O}$ is satisfiable.
- ABox retrieval: Compute all ABox axioms that are entailed by $\mathcal{O}$.
- Classification: Compute all axioms of the form $A \sqsubseteq B$ with $A, B \in N_C$ that are entailed by $\mathcal{O}$.

$$
\begin{array}{rcl}
\prod_{i=1}^{n} A_i \sqsubseteq B & \mapsto & \bigwedge_{i=1}^{n} A_i(x) \rightarrow B(x) \\
A \sqsubseteq \exists R.B & \mapsto & A(x) \rightarrow \exists y(R(x,y) \wedge B(y)) \\
\exists R.A \sqsubseteq B & \mapsto & R(x,y) \wedge A(y) \rightarrow B(x) \\
A \sqsubseteq \forall R.B & \mapsto & A(x) \wedge R(x,y) \rightarrow B(y) \\
A \sqsubseteq \leq 1\, R.B & \mapsto & A(x) \wedge R(x,y) \wedge B(y) \wedge R(x,z) \wedge B(z) \rightarrow y \approx z \\
A \sqsubseteq \{a\} & \mapsto & A(x) \rightarrow x \approx a \\
A \sqsubseteq \exists R.\{a\} & \mapsto & A(x) \rightarrow R(x,a) \\
R \sqsubseteq S & \mapsto & R(x,y) \rightarrow S(x,y) \\
R^- \sqsubseteq S & \mapsto & R(x,y) \rightarrow S(y,x) \\
R \circ S \sqsubseteq V & \mapsto & R(x,y) \wedge S(y,z) \rightarrow V(x,z)
\end{array}
$$

Figure 3.2: Function π Mapping DL Axioms to Rules. In the above, $A_{(i)}, B \in N_C$, $R, S, V \in N_R$, $a \in N_I \cap \mathbf{C}$ and $x, y, z \in \mathbf{V}$.

3.3 The Relation between DLs and Existential Rules

In subsequent chapters, we employ mappings from ontologies to programs in such a way that the latter can be used to solve reasoning tasks over the former. In this section, we briefly introduce one of such mappings which is reused across further sections in this document.

Definition 3.3.1 *Given a TBox $\mathcal{T}$, let $\mathcal{R}(\mathcal{T}) = \{\pi(\alpha) \mid \alpha \in \mathcal{T}\}$ where π is the function from Figure 3.2. Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, let $\mathcal{P}(\mathcal{O}) = \langle \mathcal{R}(\mathcal{T}), \mathcal{A} \rangle$.*

3.4 Axiomatizations of the Equality Predicate

In FOL, the equality predicate is commonly assumed to have a predefined interpretation. Nevertheless, equality can also be treated as an ordinary predicate with an explicit axiomatization. In this section, we introduce two different approaches to axiomatize this predicate.

3.4.1 The Standard Equality Axiomatization

Definition 3.4.1 *Given a formula ϕ, let $\mathsf{Eq}(\phi)$ be the formula that results from substituting every atom of the form $t \approx u$ by $\mathsf{Eq}(t,u)$ where Eq is a fresh predicate. Given a program $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$, let $\mathsf{Eq}(\mathcal{P}) = \langle \mathsf{Eq}(\mathcal{R}), \mathcal{I} \rangle$.*

Let $\mathcal{P}$ be some program. Then, the standard equality axiomatization *of $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$ is the program $\mathsf{SEA}(\mathcal{P}) = \langle \mathcal{R}', \mathcal{I} \rangle$ where $\mathcal{R}'$ is the set of rules containing the set $\mathsf{Eq}(\mathcal{R})$, the rules (3.15-3.17) from Figure 3.3, and an instance of rule (3.18) for every n-ary predicate p in $\mathcal{P}$ for every $1 \leq i \leq n$.*

As shown by the following result, the standard equality axiomatization preserves CQ entailment.

$$\top(x) \rightarrow \mathsf{Eq}(x, x) \tag{3.15}$$

$$\mathsf{Eq}(y, x) \rightarrow \mathsf{Eq}(x, y) \tag{3.16}$$

$$\mathsf{Eq}(x, z) \wedge \mathsf{Eq}(z, y) \rightarrow \mathsf{Eq}(x, y) \tag{3.17}$$

$$p(x_1, \ldots, x_i, \ldots, x_n) \wedge \mathsf{Eq}(x_i, x_i') \rightarrow p(x_1, \ldots, x_i', \ldots, x_n) \tag{3.18}$$

Figure 3.3: Rules Employed in the Axiomatization of Equality.

Proposition 3.4.2 *Given a program $\mathcal{P}$ and a query γ, $\mathcal{P} \models \gamma$ if and only if $\mathsf{SEA}(\mathcal{P}) \models \gamma$.*

3.4.2 Singularization

Another method to axiomatize the meaning of equality is *singularization*, first described in [Marnette 2009].

Definition 3.4.3 *A* singularization of a conjunct of atoms β *is a conjunct of atoms that results from performing the following transformation to β: For every variable x occurring in β, (i) arbitrarily select some occurrence of x, (ii) replace each other occurrences of x with different fresh variables $x_1, \ldots, x_n$ and, (iii) for every $i = 1, \ldots, n$, add the conjunct $Eq(x, x_i)$ to β, where Eq is a fresh predicate.*

A singularization of a rule ρ *is a rule that results from substituting the body of ρ by some of its singularizations and replacing every atom of the form $t \approx u$ in the head with $Eq(t, u)$. A* singularization of a query $\exists\vec{x}\beta$ *is a query with a singularization of β as body.*

A singularization of a set of rules $\mathcal{R}$ *is the set of rules containing rules (3.15-3.17) and exactly one singularization for every rule in $\mathcal{R}$. The* singularization of a program $\mathcal{P}$ *is a program that results from replacing the set of rules in $\mathcal{P}$ with some of its singularizations.*

Given a conjunction of atoms, query, rule, set of rules or program ϕ, let $\mathsf{Sing}(\phi)$ be the set of all possible singularizations of ϕ.

Example 3.4.4 *Let $\rho = A(x) \wedge B(x) \wedge C(x) \rightarrow \exists y(R(x, y) \wedge B(y))$. Such rule admits the following singularizations.*

$$A(x) \wedge B(x_1) \wedge C(x_2) \wedge Eq(x, x_1) \wedge Eq(x, x_2) \rightarrow \exists y(R(x, y) \wedge B(y))$$

$$A(x_1) \wedge B(x) \wedge C(x_2) \wedge Eq(x, x_1) \wedge Eq(x, x_2) \rightarrow \exists y(R(x, y) \wedge B(y))$$

$$A(x_1) \wedge B(x_2) \wedge C(x) \wedge Eq(x, x_1) \wedge Eq(x, x_2) \rightarrow \exists y(R(x, y) \wedge B(y))$$

As shown in [Marnette 2009], any singularization of a program $\mathcal{P}$ can alternatively be employed in place of $\mathcal{P}$ to solve BCQ entailment.

Proposition 3.4.5 *A program entails a query if and only if every singularization of such program entails every singularization of such query.*

3.5 The Chase Algorithm

BCQ entailment over programs and Horn ontologies can, in some cases, be addressed via application of the chase algorithm, a technique where all relevant consequences of an ontology are precomputed, allowing queries to be directly evaluated on the materialized set of facts. In this section we present two variants of the chase algorithm, which are somewhat similar to the oblivious and restricted chase from [Calì et al. 2013].

Definition 3.5.1 *A fact ϕ is an* oblivious consequence *of a TGD $\rho = \beta(\vec{x}, \vec{y}) \rightarrow \exists \vec{z} \eta(\vec{x}, \vec{z})$ on a set of facts $\mathcal{F}$ if and only if there is some substitution σ with $\beta(\vec{x}, \vec{z})\sigma \subseteq \mathcal{F}$ and $\phi \in sk(\eta(\vec{x}, \vec{y}))\sigma$ where $sk(\eta(\vec{x}, \vec{y}))$ is the head of the (skolemized) TGD $sk(\rho)$. A fact ϕ is a* restricted consequence *of ρ on $\mathcal{F}$ if and only if there is a substitution σ with (1) $\beta(\vec{x}, \vec{z})\sigma \subseteq \mathcal{F}$ and $\phi \in sk(\eta(\vec{x}, \vec{y}))\sigma$, and (2) there is no substitution $\tau \supseteq \sigma$ with $\eta(\vec{x}, \vec{y})\tau \subseteq \mathcal{F}$.*

The result of obliviously applying *ρ to $\mathcal{F}$, written $\rho_O(\mathcal{F})$, is the set of all* oblivious consequences *of ρ on $\mathcal{F}$. The result of* obliviously applying *a set of TGDs $\mathcal{R}$ to $\mathcal{F}$, written $\mathcal{R}_O(\mathcal{F})$, is the set $\bigcup_{\rho \in \mathcal{R}} \rho_O(\mathcal{F}) \cup \mathcal{F}$. The result of* restrictively applying *ρ to $\mathcal{F}$ (resp., $\mathcal{R}$ to $\mathcal{F}$), written $\rho_R(\mathcal{T})$ (resp., $\mathcal{R}_R(\mathcal{T})$), is analogously defined.*

Definition 3.5.2 *Let $\rightsquigarrow$ be some total strict order over the set of all terms such that $t \rightsquigarrow u$ only if $dep(t) \leq dep(u)$. Furthermore, we say that t is greater than u with respect to $\rightsquigarrow$ to indicate $t \rightsquigarrow u$.*

Given a set of EGDs $\mathcal{R}$ and a set of facts $\mathcal{F}$, let $\mapsto^{\mathcal{R}}_{\mathcal{F}}$ be the minimal congruence relation over terms such that $t \mapsto^{\mathcal{R}}_{\mathcal{F}} u$ if and only if there exists some $\beta(\vec{x}) \rightarrow x \approx y \in \mathcal{R}$ and some substitution σ with $\beta(\vec{x})\sigma \subseteq \mathcal{F}$, $\sigma(x) = t$ and $\sigma(y) = u$. Let $\mathcal{R}(\mathcal{F})$ be the set that is obtained from $\mathcal{F}$ by replacing all occurrences of every term t by u where u is the greatest term with respect to $\rightsquigarrow$ such that $t \mapsto^{\mathcal{R}}_{\mathcal{F}} u$.

Note that we define consequences with respect to sets of rules instead of simply (single) rules as it is customary [Calì et al. 2013]. This allows us to define the chase as a deterministic procedure (modulo $\rightsquigarrow$). Also, unlike in [Calì et al. 2013], where a lexicographic order is used to direct the replacement of terms, we employ a type of order which ensures that terms are always replaced by terms of equal or lesser depth. This effectively precludes some terms with larger depth from being introduced during the computation of the chase.

Definition 3.5.3 *Let $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$ be some program. The* oblivious chase sequence *of $\mathcal{P}$ is the sequence $\mathcal{F}_0, \mathcal{F}_1, \ldots$ such that $\mathcal{F}_1 = \mathcal{I}$ and, for all $i \geq 1$, $\mathcal{F}_i$ is the set of facts defined as follows.*

- *If $\mathcal{R}^{\approx}(\mathcal{F}_{i-1}) \neq \mathcal{F}_{i-1}$, then $\mathcal{F}_i = \mathcal{R}^{\approx}(\mathcal{F}_{i-1})$.*
- *If $\mathcal{F}_{i-1} = \mathcal{R}^{\approx}(\mathcal{F}_{i-1})$ and $\mathcal{F}_{i-1} \neq \mathcal{R}^{\forall}_O(\mathcal{F}_{i-1})$, then $\mathcal{F}_i = \mathcal{R}^{\forall}_O(\mathcal{F}_{i-1})$.*
- *Otherwise, $\mathcal{F}_i = \mathcal{R}^{\exists}_O(\mathcal{F}_{i-1})$.*

The restricted chase sequence *of $\mathcal{P}$ is defined analogously.*

For the sake of brevity, we frequently denote the oblivious (resp., restricted) chase sequence of a program $\mathcal{P}$ with $\mathcal{P}^1_O, \mathcal{P}^2_O, \ldots$ (resp., $\mathcal{P}^1_R, \mathcal{P}^2_R, \ldots$).

$$
\begin{aligned}
\mathcal{T} &= \{Film \sqsubseteq \exists isProdBy.Producer, Producer \sqsubseteq \exists prod.Film, isProdBy^{-} \sqsubseteq prod,\\
&\qquad prod^{-} \sqsubseteq isProdBy\}\\
\mathcal{O} &= \langle \mathcal{T}, \{Film(AI)\}\rangle\\
\mathcal{R}(\mathcal{T}) &= \{\rho = Film(x) \rightarrow \exists y[isProdBy(x,y) \wedge Producer(y)],\\
&\qquad \upsilon = Producer(x) \rightarrow \exists y[prod(x,y) \wedge Film(y)],\\
&\qquad isProdBy(y,x) \rightarrow prod(x,y), prod(y,x) \rightarrow isProdBy(x,y)\}\\
\mathcal{P}(\mathcal{O}) &= \langle \mathcal{R}(\mathcal{T}), \{Film(AI)\}\rangle\\
\mathcal{P}(\mathcal{O})^1_R &= \{Film(AI), isProdBy(AI, f^y_\rho(AI)), Producer(f^y_\rho(AI))\}\\
\mathcal{P}(\mathcal{O})^2_R &= \{prod(f^y_\rho(AI), AI)\} \cup \mathcal{P}(\mathcal{O})^1_O\\
RC(\mathcal{P}(\mathcal{O})) &= \mathcal{P}(\mathcal{O})^2_O\\
OC(\mathcal{P}(\mathcal{O})) &= RC(\mathcal{P}(\mathcal{O})) \cup \{prod(f^y_\rho(AI), f^y_\upsilon(f^y_\rho(AI))), Film(f^y_\upsilon(f^y_\rho(AI))), \ldots\}
\end{aligned}
$$

Figure 3.4: Ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A}\rangle$, Program $\mathcal{P}(\mathcal{O})$ and the (Restricted and Oblivious) Chase of $\mathcal{P}(\mathcal{O})$.

Definition 3.5.4 *Let $\mathcal{P}$ be some program and let $\mathcal{R}$ be some set of rules. Then, the* oblivious chase *of $\mathcal{P}$ is the set $OC(\mathcal{P}) = \bigcup_{i\in\mathbb{N}} \mathcal{P}^i_O$. The* restricted chase *of $\mathcal{P}$, written $RC(\mathcal{P})$, is defined analogously.*

The oblivious (resp., restricted) chase of $\mathcal{P}$ terminates *if and only if there is some i such that, for all $j \geq i$, $\mathcal{P}^i_O = \mathcal{P}^j_O$. Furthermore, the oblivious (resp., restricted) chase of a set of rules $\mathcal{R}$* terminates *if the oblivious (resp., restricted) chase of every program of the form $\langle \mathcal{R}, \mathcal{I}\rangle$ terminates.*

The (restricted or oblivious) chase of a program can be employed to solve CQ entailment [Calì et al. 2013]. I.e., a program $\mathcal{P}$ entails a query γ, written $\mathcal{P} \models \gamma$, if and only if either $OC(\mathcal{P}) \models \exists y \bot(y)$ or $OC(\mathcal{P}) \models \gamma$ (resp., $RC(\mathcal{P}) \models \exists y \bot(y)$ or $RC(\mathcal{P}) \models \gamma$). Thus, we may also use the chase to solve CQ entailment over ontologies: An ontology $\mathcal{O}$ entails a query γ if and only if $OC(\mathcal{P}(\mathcal{O})) \models \exists y \bot(y)$ or $OC(\mathcal{P}(\mathcal{O})) \models \gamma$ (resp., $RC(\mathcal{P}(\mathcal{O})) \models \exists y \bot(y)$ or $RC(\mathcal{P}(\mathcal{O})) \models \gamma$).

In the above paragraph, $\models$ represents the standard FOL entailment relationship. As customary, given a set of facts $\mathcal{F}$ and a query $\gamma = \exists \vec{x}\beta$, $\mathcal{F} \models \gamma$ if and only if there is some substitution σ such that $\beta\sigma \subseteq \mathcal{F}$.

For readability purposes, we say that the oblivious (resp. restricted) chase of some ontology $\mathcal{O}$ *terminates* if and only if the oblivious (resp. restricted) chase of $\mathcal{P}(\mathcal{O})$ terminates. The oblivious (resp. restricted) chase of some TBox $\mathcal{T}$ *terminates* if and only if if the oblivious (resp. restricted) chase of $\mathcal{R}(\mathcal{T})$ terminates.

The restricted chase has a relevant advantage over the oblivious chase: in some cases, the former might terminate whereas the latter does not.

Example 3.5.5 *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A}\rangle$ be as in Figure 3.4. This figure depicts the oblivious and restricted chase sequence of $\mathcal{P}(\mathcal{O})$. In this case, $RC(\mathcal{P}(\mathcal{O}))$ terminates whilst $OC(\mathcal{P}(\mathcal{O}))$ does not.*

4

The RCA_n Fragment of Horn DL

In some cases, CQ answering over Horn DL can be addressed via application of the chase algorithm (see Section 3.5 for a thorough description of this reasoning algorithm). However, the chase is not guaranteed to terminate for all ontologies, and checking whether it does is not a straightforward procedure. It is thus an ongoing research endeavor to establish so-called *acyclicity conditions*; i.e., sufficient conditions which characterize ontologies for which the chase does indeed terminate.

The main contribution presented in this chapter is the definition of *restricted chase acyclicity* (RCA_n), a novel acyclicity condition applicable to Horn-$\mathcal{SRIQ}$ ontologies. If an ontology is proven to be RCA_n, then n-cyclic terms do not occur during the computation of the chase of such ontology and thus, the chase is guaranteed to terminate. Note that, an acyclicity notion such as RCA_n may be regarded as the definition of a language; i.e., the RCA_n fragment of Horn DL is the set of all Horn ontologies that are RCA_n.

In contrast with existing acyclicity notions [Cuenca Grau et al. 2013] which deal with termination of the oblivious chase of arbitrary sets of existential rules, we restrict our attention to the language Horn-$\mathcal{SRIQ}$ and seek to achieve termination of the restricted chase algorithm. As described in Section 3.5, this is a special variant of the chase in which the inclusion of further terms to satisfy existential restrictions is avoided if such restrictions are already satisfied. By considering such a chase algorithm we are able to devise acyclicity conditions which are empirically more general than any other of the notions previously described.

On the theoretical side, we show that RCA_n is more general than *model-faithful acyclicity* (MFA) provided n is linearly large size of ontology. As shown in [Cuenca Grau et al. 2013], MFA is one of the most general acyclicity conditions for ontologies described to date, as it encompasses many other existing notions such as *joint acyclicity* [Krötzsch and Rudolph 2011], *super-weak acyclicity* [Marnette 2009] or the hybrid acyclicity notions presented in [Baget et al. 2014]. Furthermore, we show that deciding RCA_n membership of an ontology is not harder than deciding MFA membership.

On the practical side, we empirically show that (i) RCA_n characterizes more real-world ontologies as acyclic than MFA. Furthermore, we demonstrate that (ii) the use of RCA_n results in a more efficient reasoning procedure. This is because acyclicity is still preserved in the case when employing renaming techniques when reasoning in the presence of equality. Thus, the use of cumbersome axiomatizations

of equality such as *singularization* [Marnette 2009] can be avoided. Moreover, we report on an implementation of the restricted chase algorithm based on the datalog engine RDFOx [Nenov et al. 2015] and show that (iii) it vastly outperforms state-of-the-art DL reasoners. To verify (i-iii), we complete an extensive evaluation with very encouraging results.

4.1 Model Faithful Acyclicity

In this section we briefly describe MFA [Cuenca Grau et al. 2013], one of the most general acyclicity notions for sets of existential rules. MFA guarantees the termination of the oblivious chase of a program by precluding the occurrence of cyclic terms in the chase.

When one is interested in checking the termination of the oblivious chase of a set of rules with respect to every possible instance, it is enough to check termination with respect to a special instance, the *critical instance* [Marnette 2009]. The critical instance is the minimal set which contains all possible atoms that can be formed using the relational symbols which occur in TGDs and the special constant $\star$. Such a strategy is used by MFA to guarantee termination of a set of rules.

Definition 4.1.1 *The* critical instance $\mathcal{I}_\star(\mathcal{R})$ *of a set of rules* $\mathcal{R}$ *is the set of all facts that can be constructed using relational symbols in* $\mathcal{R}$ *and the fresh constant* $\star$*.*

Definition 4.1.2 *A set or TGDs* $\mathcal{R}$ *is MFA if and only if no cyclic term occurs in the oblivious chase sequence of the program* $\langle \mathcal{R}, \mathcal{I}_\star(\mathcal{R}) \rangle$*.*

The previous definition may also be applied to set of axioms in the following manner. A TBox $\mathcal{T}$ is MFA if and only if the set $\mathcal{R}(\mathcal{T})$ is MFA (the set $\mathcal{R}(\mathcal{T})$ is introduced in Definition 3.3.1). Note that, a condition such as MFA can be applied to check whether a TBox $\mathcal{T}$ is acyclic; i.e., $\mathcal{T}$ is MFA if and only if $\mathcal{R}(\mathcal{T})$ is MFA.

While the actual definition of MFA does not preclude the existence of EGDs, equality is assumed to be axiomatized, and thus it is treated as a regular predicate (EGDs are de facto TGDs). Nevertheless, the use of the standard equality approach to axiomatize equality tends to make the MFA membership check fail (see [Cuenca Grau et al. 2013] for a lengthier discussion on this topic). Thus, the use of singularization [Marnette 2009] (introduced in Section 3.4.2), a somewhat "less-harmful" axiomatization of equality, is proposed in [Cuenca Grau et al. 2013]. The use of singularization results in the definition of several variants of MFA, which are introduced in the following definition.

Definition 4.1.3 *For a set of TGDs* $\mathcal{R}$*, if there is some set of rules in* $\mathsf{Sing}(\mathcal{R})$ *which is MFA, then* $\mathcal{R}$ *is MFA*$^\exists$*. If every set in* $\mathsf{Sing}(\mathcal{R})$ *is MFA, then* $\mathcal{R}$ *is MFA*$^\forall$*.*

Due to the high number of possible singularizations, it is frequently not feasible to check MFA$^\exists$ or MFA$^\forall$ membership. Given a set of TGDs $\mathcal{R}$, a simpler alternative is to check whether $\bigcup_{\mathcal{R}' \in \mathsf{Sing}(\mathcal{R})} \mathcal{R}'$ is MFA. If that is the case, then $\mathcal{R}$ is said to be MFA$^\cup$. Note that given some Horn-$\mathcal{SRIQ}$ TBox $\mathcal{T}$, the set $|\bigcup_{\mathcal{R}' \in \mathsf{Sing}(\mathcal{R}(\mathcal{T}))} \mathcal{R}'|$ is actually polynomial in $|\mathcal{T}|$ and, as such, MFA$^\cup$ is more feasible to check. Thus, we will use MFA$^\cup$ as a baseline for the empirical evaluation of our novel acyclicity condition RCA$_n$, which is introduced in the next section.

4.2 Restricted Chase Acyclicity

While MFA is quite a general acyclicity condition, it has two main drawbacks:

1. It only considers the oblivious chase, which as we have seen in Example 3.5.5, might not terminate (even though the restricted chase does!), and
2. its treatment of equality via singularization is cumbersome and inefficient in practice. Not only $\text{MFA}^{\exists}$ and $\text{MFA}^{\forall}$ are difficult to check, but even after a set of TGDs are established to belong to some MFA subclass, one has to employ a singularized program for reasoning purposes.

In this section, we present RCA_n, an acyclicity notion with neither of these drawbacks: RCA_n verifies termination of the restricted chase of a TBox and does not require the use of cumbersome axiomatizations of the equality predicate.

Since we are primarily interested in termination of the restricted chase of a Horn-$\mathcal{SRIQ}$ TBox, one might wonder why we do not simply check for termination of the restricted chase for such a TBox with respect to the critical instance, as it is done in the previous section with the oblivious chase. Unfortunately, this is not possible: The restricted chase of any set of existential rules always terminates with respect to the critical instance. Thus, we have to devise more sophisticated techniques to check the termination of the restricted chase. We start by introducing the notion of an overchase for a TBox.

Definition 4.2.1 *A set of facts $\mathcal{V}$ is an* overchase *for some TBox $\mathcal{T}$ if and only if, for every $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, $RC(\mathcal{P}(\mathcal{O}))_\star \subseteq \mathcal{V}$.*

As defined in Chapter 3, the set $RC(\mathcal{P}(\mathcal{O}))_\star$ is the set that results from substituting every syntactic occurrence of constant in $RC(\mathcal{P}(\mathcal{O}))$ by $\star$. Given some TBox $\mathcal{T}$, an overchase for $\mathcal{T}$ may be intuitively regarded as an over-approximation of the restricted chase of $\mathcal{T}$.

Lemma 4.2.2 *If there exists a finite overchase for a TBox, then the restricted chase of such TBox terminates.*

Proof 1 *Let $\mathcal{V}$ be some finite overchase of $\mathcal{T}$. By the definition of an overchase, we have that, for every ontology of the form $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, $|RC(\mathcal{P}(\mathcal{O}))|\ |N_I(\mathcal{O})|^2 \leq |\mathcal{V}|$ where $N_I(\mathcal{O})$ is the set of all individuals in $\mathcal{O}$. Note that, the maximum arity of a fact occurring in $RC(\mathcal{P}(\mathcal{O}))$ is 2. By Lemma 4.4.2, we have that, once a fact ϕ is removed from the chase sequence, then ϕ may not reoccur. Thus, having an upper bound on the number of facts that may occur in $RC(\mathcal{O})$ for every program of the form $\mathcal{P}(\mathcal{O})$ where $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, termination of the restricted chase of $\mathcal{T}$ is guaranteed.*

Thus, to determine whether the chase of a TBox $\mathcal{T}$ terminates, we introduce a procedure to compute an overchase for $\mathcal{T}$ and a means to check its termination. We proceed with some preliminary notions and notation.

Definition 4.2.3 *Let $\mathcal{T}$ be some TBox and t a term. Let $\mathcal{I}(t)$ be the set of facts defined as follows: If t is of the form $f^y_\rho(s)$ where $\rho = A(x) \to \exists y[R(x,y) \wedge B(y)]$, then $\mathcal{I}(t) = \{A(s), R(s,t), B(t)\} \cup \mathcal{I}(s)$; otherwise, $\mathcal{I}(t) = \emptyset$. Furthermore, we introduce the program $\mathcal{U}(\mathcal{T}, t) = \langle \mathcal{R}(\mathcal{T})^\forall \cup \mathcal{R}(\mathcal{T})^\approx, \mathcal{I}(t)\rangle$.*

Intuitively, the restricted chase of the program $\mathcal{U}(\mathcal{T}, t)$ can be regarded as some kind of under-approximation of the facts that must occur in the chase of every program of the form $\mathcal{P}(\langle \mathcal{T}, \mathcal{A}\rangle)$ where t occurs. I.e., if t occurs in the restricted chase sequence of any program $\mathcal{P}(\langle \mathcal{T}, \mathcal{A}\rangle)$, then the facts in the restricted chase of $\mathcal{U}(\mathcal{T}, t)$ must also occur (up to renaming) in the chase sequence of such program. Furthermore, due to the special priority of application of the rules during the computation of the chase, the facts in the restricted chase of $\mathcal{U}(\mathcal{T}, t)$ must occur in the restricted chase sequence of every program of the form $\mathcal{P}(\langle \mathcal{T}, \mathcal{A}\rangle)$ before any successors of t are introduced.

Example 4.2.4 *Let $\mathcal{O}$, ρ and υ be the ontology and rules from Example 3.5.5. Then, by Definition 4.2.3:*

$$\mathcal{I}(f^y_\rho(AI)) = \{Film(AI), isProdBy(AI, f^y_\rho(AI)), Producer(f^y_\rho(AI))\} \text{ and}$$
$$RC(\mathcal{U}(\mathcal{T}, f^y_\rho(AI))) = \{prod(f^y_\rho(AI), AI)\} \cup \mathcal{I}(f^y_\rho(AI)).$$

All the facts in the restricted chase of $\mathcal{U}(\mathcal{T}, t)$ occur in the restricted chase sequence of $\mathcal{P}(\mathcal{O})$ before any successors of term $f^y_\rho(AI)$ are introduced. This is because the rule $isProdBy(y, x) \to prod(x, y)$ is applied with a higher priority than the rule $\upsilon = Producer(x) \to \exists y[prod(x, y) \wedge Film(y)]$.

Given a TBox $\mathcal{T}$ and some term of the form $f^y_\rho(t)$, we can in some cases conclude that such a term may never occur during the computation of the restricted chase of every program of the form $\mathcal{P}(\langle \mathcal{T}, \mathcal{A}\rangle)$ by carefully inspecting the facts in the set $\mathcal{U}(\mathcal{T}, t)$.

Definition 4.2.5 *Let $\mathcal{T}$ be a TBox and t a term of the form $f^y_\rho(s)$ where $\rho = A(x) \to \exists y[R(x,y) \wedge B(y)]$. We say that a term t is* restricted with respect to $\mathcal{T}$ *if and only if there is some term u with $\{R([s], u), B(u)\} \subseteq RC(\mathcal{U}(\mathcal{T}, s))$ where $[s] = [v]$, if s is replaced by v during the computation of the restricted chase sequence; and $[s] = s$, otherwise.*

We often simply say that a term is "restricted", instead of "restricted with respect to $\mathcal{T}$," if the TBox $\mathcal{T}$ is clear from the context.

Lemma 4.2.6 *Let $\mathcal{T}$ be a TBox and t a restricted term. Then, for every possible $\mathcal{O} = \langle \mathcal{T}, \mathcal{A}\rangle$, $t \notin RC(\mathcal{P}(\mathcal{O}))$.*

Proof 2 *(Sketch) Let t be a term of the form $f^y_\rho(s)$ where $\rho = A(x) \to \exists y(R(x,y) \wedge B(y))$. We can verify that, if t occurs during the computation of the chase sequence, then every fact $RC(\mathcal{U}(\mathcal{T}, s))$ will also be included in such chase sequence before any new terms are introduced. Thus, if t is indeed restricted, there must be some u with $R([s], u)$ and $B(u)$ occurring in the chase sequence. Therefore, by the definition of the chase, the term t may never be derived.*

For a complete proof of the previous lemma, see Section 4.4.2.

$\forall$-rule	if	there is some TGD of the form $\rho = \beta(\vec{x}, \vec{y}) \rightarrow \eta(\vec{x}) \in \mathcal{R}(\mathcal{T})$
	then	$\mathcal{V}_\mathcal{T} \rightarrow \rho_R(\mathcal{V}_\mathcal{T}) \cup \mathcal{V}_\mathcal{T}$
$\exists$-rule	if	there is some TGD of the form $\rho = A(x) \rightarrow \exists y[R(x, y) \wedge B(y)] \in \mathcal{R}(\mathcal{T})$ and there exists some substitution σ such that (i) $A(x)\sigma \subseteq \mathcal{V}_\mathcal{T}$ and (ii) $f^y_\rho(x)\sigma$ is not restricted with respect to $\mathcal{T}$
	then	$\mathcal{V}_\mathcal{T} \rightarrow \{R(x, f^y_\rho(x)), B(f^y_\rho(x))\}\sigma \cup \mathcal{V}_\mathcal{T}$
$\approx$-rule	if	there is some EGD $\beta(\vec{x}, \vec{y}) \rightarrow x \approx y \in \mathcal{R}(\mathcal{T})$ and there exists some substitution σ such that $\beta(\vec{x}, \vec{y})\sigma \subseteq \mathcal{V}_\mathcal{T}$
	then	$\mathcal{V}_\mathcal{T} \rightarrow \{\mathsf{Eq}(x, y), \mathsf{Eq}(y, x)\}\sigma \cup \mathcal{V}_\mathcal{T}$
Eq-rule	if	there are some terms t, u and u_i where $i = 1, \ldots, n$ and some predicate p such that (i) $p \neq \mathsf{Eq}$, (ii) $\{\mathsf{Eq}(t, u), p(u_1, \ldots, u_n)\} \subseteq \mathcal{V}_\mathcal{T}$, (iii) $dep(t) \leq dep(u)$ and (iv) $u = u_j$ for some $j = 1, \ldots, n$
	then	$\mathcal{V}_\mathcal{T} \rightarrow \{p(u_1, \ldots, u_n)\}[u/t] \cup \mathcal{V}_\mathcal{T}$

Figure 4.1: Expansion rules for the construction of $\mathcal{V}_\mathcal{T}$.

Example 4.2.7 *Let $\mathcal{T}$, ρ and υ be the TBox and rules from Example 3.5.5. We proceed to show that the term $f^y_\rho(f^y_\upsilon(AI))$ is restricted. First, we compute the restricted chase of $\mathcal{U}(\mathcal{T}, f^y_\upsilon(AI))$.*

$$RC(\mathcal{U}(\mathcal{T}, f^y_\upsilon(AI))) = \{Producer(AI), prod(AI, f^y_\upsilon(AI)), Film(f^y_\upsilon(AI)), isProdBy(f^y_\upsilon(AI), AI)\}$$

Note that $\{isProdBy(f^y_\upsilon(AI), AI), Producer(AI)\} \subseteq RC(\mathcal{U}(\mathcal{T}, f^y_\upsilon(AI)))$. Thus, $f^y_\rho(f^y_\upsilon(AI))$ is restricted with respect to $\mathcal{T}$ and, by Lemma 4.2.6, it may not occur in the restricted chase of a program of the form $\mathcal{P}(\langle\mathcal{T}, \mathcal{A}\rangle)$. Furthermore, by Definition 4.2.5, if $f^y_\rho(f^y_\upsilon(AI))$ is restricted, then every term of the form $f^y_\rho(f^y_\upsilon(c))$, where c is a constant, is also restricted.

With Definition 4.2.5 and Lemma 4.2.6 in place, we proceed with the definition of a procedure to construct an overchase for some given TBox $\mathcal{T}$.

Definition 4.2.8 *Let $\mathcal{T}$ be a TBox. We define $\mathcal{V}_\mathcal{T}$ as the set initially containing every fact in $\mathcal{I}_\star(\mathcal{R}(\mathcal{T}))$ which is then expanded by exhaustively applying the rules in Figure 4.1 (in non-deterministic order).*

Lemma 4.2.9 *The set $\mathcal{V}_\mathcal{T}$ is an overchase of the TBox $\mathcal{T}$.*

Proof 3 *(Sketch) The lemma can be proven via induction on chase sequence of any ontology of the form $\mathcal{O} = \langle\mathcal{T}, \mathcal{A}\rangle$. Note that, $\mathcal{O}^0_R \subseteq \mathcal{V}_\mathcal{T}$ by the definition of $\mathcal{V}_\mathcal{T}$. It can be verified that, for every possible derivation of a set of facts during the computation of the chase of $\mathcal{O}$, such facts will always be contained in $\mathcal{V}_\mathcal{T}$.*

See Section 4.4.3 for a complete proof of the previous lemma.

Corollary 4.2.10 *The restricted chase of some TBox $\mathcal{T}$ terminates if $\mathcal{V}_\mathcal{T}$ is finite.*

Example 4.2.11 *Let $\mathcal{T}$ be the TBox from Example 3.5.5. Then $\mathcal{V}_\mathcal{T}$ is as follows.*

$$\mathcal{V}_\mathcal{T} = \{Film(\star), isProdBy(\star), Producer(\star), prod(\star,\star),$$
$$isProdBy(\star, f^y_\rho(\star)), Producer(f^y_\rho(\star)), prod(\star, f^y_\upsilon(\star)), Producer(f^y_\upsilon(\star))\}$$

Note that terms $f^y_\rho(f^y_\upsilon(\star))$ and $f^y_\upsilon(f^y_\rho(\star))$ are restricted and thus, they are not included in $\mathcal{V}_\mathcal{T}$. Since $\mathcal{V}_\mathcal{T}$ is finite, we can conclude termination of the restricted chase of the TBox $\mathcal{T}$.

In the previous example, we were able to ascertain termination of the restricted chase of $\mathcal{T}$ after verifying that the set $\mathcal{V}_\mathcal{T}$ is finite. A sufficient condition for finiteness of $\mathcal{V}_\mathcal{T}$ is to only allow cyclic terms up to a certain depth in this set. We use such condition to formally define RCA_n.

Definition 4.2.12 *A TBox $\mathcal{T}$* is RCA_n *if and only if there are no n-cyclic terms in $\mathcal{V}_\mathcal{T}$. An ontology $\langle\mathcal{T},\mathcal{A}\rangle$ is RCA_n if and only if $\mathcal{T}$ is RCA_n.*

Theorem 4.2.13 *If a TBox $\mathcal{T}$ is RCA_n then the restricted chase of $\mathcal{T}$ terminates.*

We proceed with several results regarding the complexity of deciding RCA_n membership and reasoning over RCA_n ontologies.

Theorem 4.2.14 *Deciding whether some TBox $\mathcal{T}$ is RCA_n is in* ExpTime.

Theorem 4.2.15 *Let $\mathcal{O} = \langle\mathcal{T},\mathcal{A}\rangle$ be some RCA_n ontology and γ a query. Then, checking whether $\mathcal{O} \models \gamma$ is* ExpTime*-complete.*

The complete proofs of the previous results can be found in Section 4.4.4.

To close the section, we present several results in which we theoretically compare the generality of RCA_n to $\text{MFA}^\cup$.

Theorem 4.2.16 *$MFA^\cup$ does not cover RCA_1.*

Proof 4 *The TBox $\mathcal{T}$ from Example 3.5.5 is RCA_1 but not $MFA^\cup$.*

Theorem 4.2.17 *If $\mathcal{T}$ is $MFA^\cup$ then $\mathcal{T}$ is RCA_n for every $n > |\mathcal{T}^\exists|$ where $\mathcal{T}^\exists$ is the set of all existential axioms in $\mathcal{T}$.*

The proof of the previous result can be found in Section 4.4.5.

4.3 Evaluation

4.3.1 An Empirical Comparison of RCA_n and $\text{MFA}^\cup$

In this section we include an empirical comparison of the generality of RCA_n and $\text{MFA}^\cup$. For our experiments, we use the TBoxes of the ontologies in the OWL Reasoner Evaluation workshop (ORE, `https://www.w3.org/community/owled/ore-2015-workshop/`) and Ontology Design Patterns (ODP, `http://www.ontologydesignpatterns.org`) datasets. The former is a large repository

used in the ORE competition containing a large corpus of ontologies. The latter contains a wide range of smaller ontologies that capture design patterns commonly used in ontology modeling. The ORE dataset is rather large, and thus we restrict our experiments to the 294 ontologies with the smallest number of existential axioms, while skipping the 77 ontologies with the largest number of existential axioms. The number of such axioms contained in an ontology is a useful metric to predict the "hardness" of acyclicity membership tests; i.e. running these experiments would be very time-intensive, while our results, reported below, already indicate that for such very hard TBoxes $\text{MFA}^{\cup}$ and RCA_n will likely not differ much (while they differ significantly for ontologies with a lower count of existential axioms).

Horn-$\mathcal{SRIQ}$ TBoxes which can be expressed in $\mathcal{EL}^{++}$, DL-Lite$_R$ and DLP were not considered in our experiments. This is because all DLP TBoxes are acyclic (with respect to every applicable acyclicity notion known to us), and there already exist effective algorithms and efficient implementations that solve CQ answering over $\mathcal{EL}^{++}$and DL-Lite$_R$ ontologies [Stefanoni et al. 2014] (albeit, if these do not include complex roles).

The results from our experiments are summarized in Figure 4.1. The evaluated TBoxes are sorted into brackets depending on the number of existential axioms they contain. For each bracket we provide the average number of axioms in the ontologies ("Avg. Size"), the number of ontologies ("Count"), and, for every condition "X" considered, the percentage of "X acyclic" ontologies

RCA_2 and RCA_3 turned out to be indistinguishable with respect to the TBoxes considered and thus, we limit our evaluation to RCA_n with $n \leq 3$. Our tests reveal that RCA_2 is significantly more general than $\text{MFA}^{\cup}$, particularly when it comes to TBoxes with a low count of existential axioms. However note that reasoning over ontologies with few (existential) axioms is in general not trivial: All of the ontologies considered in our materialization tests (see Figure 4.2) contain less than 20 existential axioms. For TBoxes containing from 1 to 10 existential axioms in the ORE dataset, more than half of the ontologies which are not $\text{MFA}^{\cup}$ are RCA_2. Furthermore, the 4 ontologies in the ODP dataset which are not $\text{MFA}^{\cup}$ are RCA_2. Interestingly, in both repositories we could not find any ontology that is $\text{MFA}^{\cup}$ but not RCA_1. Thus, with respect to the TBoxes in our corpus, RCA_1 already proves to be more general than $\text{MFA}^{\cup}$.

In total, we looked at 312 ontologies, 62% and 75% of which are $\text{MFA}^{\cup}$ and RCA_2, respectively. To gauge the significance of this improvement, we roughly compare these numbers with the results presented in [Cuenca Grau et al. 2013]. In that paper, the authors consider a total of 336 ontologies, of which 49%, 58% and 68% are *weakly acyclic* [Fagin et al. 2005], *jointly acyclic* [Krötzsch and Rudolph 2011] and $\text{MFA}^{\cup}$, respectively. Even though the comparison is not over the same TBoxes, we verify that the improvement in generality of our notion is in line with previous iterations of related work.

4.3.2 A Materialization Based Reasoner

We now report on an implementation of the restricted chase as defined in Section 3.5. Moreover, we also present an implementation of the oblivious chase with singularization, i.e., the chase as it must

ORE

∃-Axioms	Avg. Size	Count	MFA$^{\cup}$	RCA$_1$	RCA$_2$	RCA$_3$
1-5	175	70	70.0	87.1	92.9	92.9
6-10	219	48	58.3	83.3	83.3	83.3
11-25	916	54	83.3	85.2	91	91
26-100	521	42	54.8	59.5	61.9	61.9
101-500	1290	42	26.2	26.2	28.6	28.6
501-1922	5052	38	60.5	60.5	60.5	60.5
1-1922	1362	294	60.9	70.1	73.1	73.1

ODP

∃-Axioms	Size	Total	MFA$^{\cup}$	RCA$_1$	RCA$_2$	RCA$_3$
1-12	39	18	73.7	100.0	100.0	100.0

Table 4.1: Results for the ORE and ODP Repositories.

be used if we employ MFA$^{\cup}$ (see Section 4.1). We use the datalog engine RDFOx [Nenov et al. 2015] in both implementations.

We evaluate the performance of our chase based implementations against Konclude [Steigmiller et al. 2014], a very efficient OWL DL reasoner, and PAGOdA [Zhou et al. 2015], a hybrid approach to query answering over ontologies. PAGOdA combines a datalog reasoner with a fully-fledged OWL 2 reasoner in order to provide scalable 'pay-as-you-go' performance and is, to the best of our knowledge, the only other implementation that may solve CQ answering over Horn-$\mathcal{SRIQ}$ ontologies with completeness guarantees, albeit only in some cases. Nevertheless, PAGOdA was able to solve all the queries (that is, all of which for which it did not time-out or run out of memory) in this evaluation in a sound and complete manner.

We consider two real-world ontologies in our experiments, Reactome and Uniprot, and two standard benchmarks, LUBM and UOBM, all of which contain a large amount of ABox axioms. Axioms in these ontologies which are not expressible in Horn-$\mathcal{SRIQ}$ were pruned. Furthermore, one extra axiom had to be removed from Uniprot for it to be both MFA$^{\cup}$ and RCA$_1$ acyclic.

The results from our experiments are summarized in Figure 4.2. For each ontology, we consider four samples of the original ABox. The number of triples contained in each one of these is indicated at the beginning of each row, under the column "Triples Count." As previously mentioned, we consider four different implementations: These include the two aforementioned variants of the chase ("Restricted" and "Oblivious"), PAGOdA ("PAGOdA") and Konclude ("Konc."). For both chase based implementations, we check the time it takes to compute the chase ("C") and then the time to solve each of the four queries crafted for each ontology ("Q1-Q4"). In a similar manner, we list the time PAGOdA takes to preprocess each ontology ("P") plus the time it takes to answer the queries ("Q1-Q4"). Finally, we list the time Konclude takes to solve realization; i.e., the task of computing every fact of the form $A(a)$ entailed by an ontology (note that Konclude cannot solve arbitrary CQ

Triples	Restricted					Oblivious					PAGOdA					Konc.
Count	C	Q1-Q4				C	Q1-Q4				P	Q1-Q4				R
2.8M	10	0	0	0	0	45	0	0	TO	0	89	OM	4	1	0	75
5.1M	21	0	0	0	0	138	0	0	TO	3	147	OM	1	2	0	214
6.7M	28	0	0	0	0	1029	2	0	TO	0	203	OM	2	3	1	506
8.1M	36	37	0	0	0	TO	-	-	-	-	263	OM	2	2	6	1347
9.0M	37	0	0	0	0	OM	-	-	-	-	113	1	1	1	1	198
17.8M	72	0	0	0	0	OM	-	-	-	-	232	2	2	3	3	987
26.2M	107	0	0	0	0	OM	-	-	-	-	378	4	10	12	5	3491
33.9M	141	0	1	0	0	OM	-	-	-	-	521	6	21	21	12	TO
2.8M	8	0	0	0	1	70	0	0	0	74	51	OM	0	0	0	51
5.7M	16	0	0	0	2	158	1	1	1	154	99	OM	1	1	0	118
8.4M	26	0	0	0	3	242	1	1	2	186	142	OM	2	1	1	220
11.4M	37	1	0	0	5	341	2	2	3	311	197	OM	3	1	1	315
2.2M	11	0	0	0	0	56	0	0	0	1	61	28	0	TO	1	53
4.5M	27	2	0	0	0	133	0	0	1	2	121	60	0	TO	2	125
6.6M	42	3	1	1	0	216	1	1	2	3	186	TO	0	TO	5	292
8.9M	58	5	1	2	1	310	1	2	4	6	260	TO	0	TO	5	644

Table 4.2: Results for Reactome, Uniprot, LUBM and UOBM (sorted from top to bottom in the above table).

answering). Time-outs, indicated with "TO," were set at 1 hour for materialization and 5 minutes for queries. We make use of the acronym "OM" to indicate that an out-of-memory error occurred. Sometimes, a time-out or an out of memory error prevents us from answering the queries: Such a situation is indicated with "-." All experiments were performed on a MacBook Pro with 8GB of RAM and a 2.4 GHz Intel Core i5 processor.

For each ontology, we consider four different queries which are listed in Figure 4.2 (such figure can be found at the end of the chapter). For every ontology, the query Q1 is of the form $\exists x, y, z R(x, y) \wedge R(z, y)$ where R is an existentially quantified role occurring in the TBox. It appears that PAGOdA has trouble with this kind of query, whereas the chase based implementations efficiently solve it in all but one case. This is probably due to the design of the hybrid reasoner which considers under and over approximations to provide complete answers to CQ: It appears that queries as the one previously considered find a large number of matches in the upper bound which slows down the performance of this reasoner. Queries Q2, and Q3 and Q4 are acyclic and cyclic, respectively (a query is acyclic if the shape of its body is acyclic). Even though it is well-known that answering acyclic CQs can be reduced to satisfiability [Carral and Hitzler 2012], we included such a type of query in our evaluation in an attempt to verify whether solving acyclic queries is simpler than cyclic queries (this is indeed the case theoretically). Nevertheless, our experiments do not reveal any significant differences.

4.4 Proofs

This section contains the complete formal proofs for all the lemmas and theorems presented across the chapter as well as several further intermediate and preliminary results.

4.4.1 Overchase and Termination

Lemma 4.4.1 *Let $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$ be some program and t be a term occurring in some $\mathcal{P}_R^i$. If there is some $\mathcal{P}_R^j$ such that $t \notin \mathcal{P}_R^j$ and $j \geq i$, then $t \notin \mathcal{P}_R^k$ for every $k \geq j$.*

Proof 5 *It is clear that the Lemma holds if t is a constant since constants may never occur in the set of consequences of a rule or a set of rules (note that, by definition, rules may not contain occurrences of constants). Thus, let us assume that t is of the form $f_\rho^y(t_1, \ldots, t_n)$ where $\rho = \beta(\vec{x}, \vec{z}) \rightarrow \exists y \eta(\vec{x}, \vec{y})$ and $\vec{x} = x_1, \ldots, x_n$. Furthermore, let l be the smallest natural number such that $t \in \mathcal{P}_R^l$.*

Note that, the term t may only be introduced in the chase sequence if it occurs as part as the set of consequences of the rule ρ with respect to some substitution σ such that, for every $i = 1, \ldots, n$, $\sigma(x_i) = t_i$.

The lemma follows from the following claim: For every $m \geq l$ and every substitution σ such that $\sigma(x_i) = t_i$ with $i = 1, \ldots, n$ and $\beta(\vec{x}, \vec{z})\sigma \subseteq \mathcal{P}_R^m$, there is some substitution $\tau \supseteq \sigma$ and $\eta(\vec{x}, \vec{y})\tau \subseteq \mathcal{P}_R^m$. Such a claim can be verified via induction on the sequence $\mathcal{P}_R^l, \mathcal{P}_R^{l+1} \ldots$

Lemma 4.4.2 *Let $\mathcal{P}$ be some program and ϕ be a fact occurring in some $\mathcal{P}_R^i$. If there is some $\mathcal{P}_R^j$ such that $\phi \notin \mathcal{P}_R^j$ and $j \geq i$, then $\phi \notin \mathcal{P}_R^k$ for every $k \geq j$.*

Proof 6 *By the definition of the chase sequence, ϕ must contain some term t that has been removed from the chase sequence due to some replacement as defined in Definition 3.5.2. Thus, the lemma holds since, Lemma 4.4.1, term t may not be reintroduced in any $\mathcal{P}_R^k$ where $k \geq j$.*

4.4.2 Restricted Terms

We introduce some preliminary function that allows us to keep track of the replacements of terms and atoms that are performed during the computation of the restricted chase.

Definition 4.4.3 *Let $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$ be a program and t be some term. Then we define the term $[t]_\mathcal{P}^i$ as follows:*

- *If the term t is replaced by some different term u during the computation of the restricted chase sequence of $\mathcal{P}$ up to the set $\mathcal{P}_R^i$, then $[t]_\mathcal{P}^i = [u]_\mathcal{P}^i$.*
- *Otherwise, $[t]_\mathcal{P}^i = t$.*

Furthermore, given some fact $\phi = p(t_1, \ldots, t_n)$ and a set of facts $\mathcal{F}$, let $[\phi]_\mathcal{P}^j = p([t_1]_\mathcal{P}^j, \ldots, [t_n]_\mathcal{P}^j)$ and $[\mathcal{F}]_\mathcal{P}^i = \{[\phi]_\mathcal{P}^i \mid \phi \in \mathcal{F}\}$.

We often write $[\phi]$ (resp., $[\mathcal{F}]$) instead of $[\phi]_\mathcal{P}^i$ (resp., $[\mathcal{F}]_\mathcal{P}^i$) if the set $\mathcal{P}_R^i$ is clear from the context.

Lemma 4.4.4 *Let $\mathcal{P}$ be some ontology and ϕ a fact occurring in the restricted chase of $\mathcal{P}$. Furthermore, let i be the smallest natural number such that $\phi \in \mathcal{P}_R^i$. Then, $[\phi]_j^{\mathcal{P}} \in \mathcal{P}_R^j$ for every $j \geq i$.*

Proof 7 *We prove the lemma via induction on the sequence $\mathcal{P}_R^i, \mathcal{P}_R^{i+1}, \ldots$ Let $\phi = p(t_1, \ldots, t, \ldots, t_n)$ and $\mathcal{P} = \langle \mathcal{R}, \mathcal{I} \rangle$.*

(Base Case) We proceed to show that $[\phi]_{\mathcal{P}}^i \in \mathcal{P}_R^i$. Note that, since i is the smallest number with $\phi \in \mathcal{P}_R^i$,

$$[\phi]_{\mathcal{P}}^i = p([t_1]_{\mathcal{P}}^i, \ldots, [t_n]_{\mathcal{P}}^i) = p(t_1, \ldots, t_n) = \phi.$$

(Induction Step) We proceed to show that $[\phi]_{\mathcal{P}}^k \in \mathcal{P}_R^k$ where $k > i$. By induction hypothesis (IH), we have that $[\phi]_{\mathcal{P}}^{k-1} = p([t_1]_{\mathcal{P}}^{k-1}, \ldots, [t_n]_{\mathcal{P}}^{k-1}) \in \mathcal{P}_R^{k-1}$. Two possible cases arise.

1. *$[\phi]_{\mathcal{P}}^{k-1} \notin \mathcal{P}_R^k$. Then $\mathcal{P}_R^k = \mathcal{R}^{\approx}(\mathcal{P}_R^{k-1})$ and some of the terms $[t_l]_{\mathcal{P}}^{k-1}$ must have been replaced by some different terms u_l. Nevertheless, $[\phi]_{\mathcal{P}}^k \in \mathcal{P}_R^k$ since, by the definition of $[\cdot]_{\mathcal{P}}^k$, we have that $[t_l]_{\mathcal{P}}^k = u_l$ for every term $[t_l]_{\mathcal{P}}^{k-1}$ that was been replaced by u_l.*

2. *$[\phi]_{\mathcal{P}}^{k-1} \in \mathcal{P}_R^k$. Then none of the terms $[t_l]_{\mathcal{P}}^{k-1}$ have been replaced. Then the lemma holds because $[\phi]_{\mathcal{P}}^{k-1} \in \mathcal{P}_R^k$ and, by definition, $[\phi]_{\mathcal{P}}^k = [\phi]_{\mathcal{P}}^{k-1}$.*

Lemma 4.4.5 *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be some ontology and let t be some term of the form $f_\rho^y(u)$ where $\rho = A(x) \rightarrow \exists y[R(x,y) \wedge B(y)]$ that occurs in the restricted chase of $\mathcal{P}(\langle \mathcal{T}, \mathcal{A} \rangle)$. Furthermore, let i be the smallest number such that $t \in \mathcal{P}(\mathcal{O})_R^i$. Then, $\{A(u), R(u,t), B(t)\} \subseteq \mathcal{P}(\mathcal{O})_R^i$.*

Proof 8 *Since t occurs in the chase of $\mathcal{P}(\mathcal{O})$, then $A(x) \rightarrow \exists y[R(x,y) \wedge B(y)] \in \mathcal{R}(\mathcal{T})$. Also, since i is the smallest number with $t \in \mathcal{P}(\mathcal{O})_R^i$, then $\mathcal{P}(\mathcal{O})_R^i = \mathcal{R}(\mathcal{T})^{\exists}(\mathcal{P}(\mathcal{O})_R^{i-1})$. Furthermore, the term t may only occur in $\mathcal{P}(\mathcal{O})_R^i$ if there is some substitution σ with $\sigma(x) = u$ and $A(u) \in \mathcal{P}(\mathcal{O})_R^{i-1}$. Thus, $\{R(x, f_\rho^y(x)), B(f_\rho^y)]\})\sigma \subseteq \mathcal{P}(\mathcal{O})_R^i$.*

Lemma 4.4.6 *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be some ontology and let t be some term occurring in the restricted chase of $\mathcal{P}(\mathcal{O})$. Furthermore, let i be the smallest number such that $t \in \mathcal{P}(\mathcal{O})_R^i$. Then, $[\mathcal{I}(t)] \subseteq \mathcal{P}(\mathcal{O})_R^i$.*

Proof 9 *Note that, since t is a term which occurs in the restricted chase of $\mathcal{P}(\mathcal{O})$, t is of the form $f_n(\ldots f_2((f_1(c)))$ where all f_i are unary function symbols and c is a constant. We prove the lemma via induction on the following sequence $c, f_1(c), f_2(f_1(c)), \ldots, f_n(\ldots f_2((f_1(c)))$.*

(Base Case) Note that, if t is a constant then $\mathcal{I}(t) = \emptyset$ and thus the lemma trivially holds.

(Induction Step) Let t be some term of the form $f_\rho^y(u)$, where $\rho = A(x) \rightarrow \exists y R(x,y) \wedge B(y)$ (note that every term which is not a constant occurring in $\mathcal{P}(\mathcal{O})$ must be of that form). By the definition of the set $\mathcal{I}(t)$, $\mathcal{I}(t) = \mathcal{I}(u) \cup \{A(u), R(u,t), B(t)\}$. Let j be the smallest number such that $u \in \mathcal{P}(\mathcal{O})_R^j$ (note that, necessarily $j \leq i$ since u is an immediate ancestor of t). By IH, $[\mathcal{I}(u)] \subseteq \mathcal{P}(\mathcal{O})_R^j$. Furthermore, by Lemma 4.4.4, $[\mathcal{I}(u)] \subseteq \mathcal{P}(\mathcal{O})_R^i$. By Lemma 4.4.5, $\{A(u), R(u,t), B(t)\} \subseteq \mathcal{P}(\mathcal{O})_R^i$. Thus, $[\{A(u), R(u,t), B(t)\}] \subseteq \mathcal{P}(\mathcal{O})_R^i$ since, by Definition 4.4.3, $[u] = u$ and $[t] = t$.

Lemma 4.4.7 *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be some ontology and let t be some term occurring in the restricted chase of $\mathcal{P}(\mathcal{O})$. Furthermore, let i be the smallest number with $t \in \mathcal{P}(\mathcal{O})^i_R$. Then, there exists some $j \geq i$ such that all of the following hold.*

1. *For every $\phi \in RC(\mathcal{U}(\mathcal{T}, t))$, we have that $[\phi]^j_{\mathcal{P}(\mathcal{O})} \in \mathcal{P}(\mathcal{O})^j_R$.*
2. *$\mathcal{P}(\mathcal{O})^j_R$ does not contain any successors of t.*

Proof 10 *For the rest of the proof, we simply write U instead of $\mathcal{U}(\mathcal{T}, t)$ and U^k, where $k \geq 0$, as a shortcut for $\mathcal{U}(\mathcal{T}, t)^k_R$. In the same manner, we simply write $\mathcal{P}$ instead of $\mathcal{P}(\mathcal{O})$ and $\mathcal{P}^k$ instead of $\mathcal{P}(\mathcal{O})^k_R$. Furthermore, given some $k, l \geq 0$, we write $S(\mathsf{U}^k, \mathcal{P}^l)$ if and only if $[\mathsf{U}^k] \subseteq \mathcal{P}^l$ and $\mathcal{P}^l$ does not contain any successors of the term t.*

The lemma follows if, for every set in the sequence $\mathsf{U}^0, \mathsf{U}^1, \mathsf{U}^2, \ldots$, there exists some chase step $\mathcal{P}^l$ with $S(\mathsf{U}^k, \mathcal{P}^l)$. We proceed to show that this is indeed the case via induction on such sequence.

(Base Case) We proceed to show that $S(\mathsf{U}^0, \mathcal{P}^i)$. First, note that, by Lemma 4.4.6, $[\mathcal{I}(t)] \subseteq \mathcal{P}^i$. It is clear that $\mathcal{P}^i$ may not contain any successors of t, since i is the smallest number with $t \in \mathcal{P}^i$.

(Induction Step) We verify that, for every $k \geq 2$, the existence of some m with $S(\mathsf{U}^{k-1}, \mathcal{P}^m)$ implies that there is some l with $S(\mathsf{U}^k, \mathcal{P}^l)$. We assume that there is some $\phi \in \mathsf{U}^k$ with $[\phi] \notin \mathcal{P}^m$ as, otherwise, $S(\mathsf{U}^k, \mathcal{P}^m)$ and the induction step holds. Furthermore, since $S(\mathsf{U}^{k-1}, \mathcal{P}^m)$, we have that $\phi \notin \mathsf{U}^{k-1}$. There exist two possible cases, which are considered along the following paragraphs: Either (i) $\mathsf{U}^k = \mathsf{U}^{\approx}(\mathsf{U}^{k-1})$ or (ii) $\mathsf{U}^k = \mathsf{U}^{\forall}(\mathsf{U}^{k-1})$ (note that, by definition, program Udoes not contain any rules with existentially quantified variables).

Let (i) $\mathsf{U}^k = \mathsf{U}^{\approx}(\mathsf{U}^{k-1})$. In this case, we verify that $S([\mathsf{U}^k], \mathcal{P}^{m+1})$.

We first show $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$. As previously argued, there is at least some fact ϕ with $\phi \in \mathsf{U}^k$, $\phi \notin \mathsf{U}^{k-1}$ and $[\phi] \notin \mathcal{P}^m$. Therefore, one of the following cases must hold:

1. *ϕ is of the form $C(t)$, there is some rule $\rho = A(x) \wedge R(x, y) \wedge B(y) \wedge R(x, z) \wedge B(z) \rightarrow x \approx y \in \mathsf{U}$ and some u and v such that $\{A(v), R(v, u), B(u), R(v, t),\ B(t),\ C(u)\} \subseteq \mathsf{U}^{k-1}$. By IH, $[\{A(v), R(v, u), B(u), R(v, t), B(t),\ C(u)\}] \subseteq \mathcal{P}^m$. Furthermore, $\rho \in \mathcal{P}$, by the definition of U. Since $[\phi] \notin \mathcal{P}^m$, $\mathcal{P}^m \neq \mathcal{P}^{\approx}(\mathcal{P}^m)$ and thus, by Definition 3.5.3, $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$.*
2. *ϕ is of the form $R(t, s)$ (resp., $R(s, t)$), $\rho = A(x) \wedge R(x, y) \wedge B(y) \wedge R(x, z) \wedge B(z) \rightarrow x \approx y \in \mathsf{U}$ and $\{A(v), R(v, u), B(u), R(v, t), B(t), S(u, s)(resp., S(s, u))\} \subseteq \mathsf{U}^{k-1}$ for some u and v. Analogous to the previous case.*

In either case, we have that $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$. Thus, the set $\mathcal{P}^{m+1}$ does not contain any successors of t, since, by IH, $\mathcal{P}^m$ contains no such terms.

We proceed to show that, $[\mathsf{U}^k] \subseteq \mathcal{P}^{m+1}$; i.e., we show that, for every $\phi \in \mathsf{U}^k$, we have that $[\phi] \in \mathcal{P}^{m+1}$. A fact $\phi \in \mathsf{U}^k$ only if one of the following cases holds.

1. *Let $\phi \in \mathsf{U}^{k-1}$. Then, by IH, $[\phi]^m_{\mathcal{P}} \in \mathcal{P}^m$. By Lemma 4.4.4, $[\phi]^{m+1}_{\mathcal{P}} \in \mathcal{P}^{m+1}$.*
2. *Let $\phi \notin \mathsf{U}^{k-1}$. Several possible cases arise.*

(a) ϕ *is of the form* $C(t)$*, there is some rule of the form* $\rho = A(x) \wedge R(x,y) \wedge B(y) \wedge R(x,z) \wedge B(z) \rightarrow x \approx y \in \mathsf{U}$ *and there are some* u *and* v *such that* $\{A(v), R(v,u), B(u), R(v,t), B(t), C(u)\} \subseteq \mathsf{U}^{k-1}$*. By IH,* $[\{A(v),\ R(v,u),\ B(u),\ R(v,t),\ B(t),\ C(u)\}] \subseteq \mathcal{P}^m$*. Then* $[C(t)] \in \mathcal{P}^{m+1}$

(b) ϕ *is of the form* $S(t,s)$ *(resp.,* $S(s,t)$*), there is some rule of the form* $\rho = A(x) \wedge R(x,y) \wedge B(y) \wedge R(x,z) \wedge B(z) \rightarrow x \approx y \in \mathsf{U}$ *and there are some* u *and* v *such that* $\{A(v), R(v,u), B(u), R(v,t), B(t), S(u,s)(resp.,S(s,u))\} \subseteq \mathsf{U}^{k-1}$*. Analogous to the previous case.*

Thus, $S([\mathsf{U}^k], \mathcal{P}^{m+1})$.

Let (ii) $\mathsf{U}^k = \mathsf{U}^{\forall}(\mathsf{U}^{k-1})$*. We proceed to show that* $\mathcal{P}^m \neq \mathcal{P}^{\forall}(\mathcal{P}^m)$*. As in case (i), there is some fact* ϕ *with* $\phi \in \mathsf{U}^k$*,* $\phi \notin \mathsf{U}^{k-1}$ $[\phi] \notin \mathcal{P}^m$*. Thus, one of the following cases must hold:*

1. ϕ *is of the form* $B(t)$ *and either of the following holds:*

   (a) $A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow B(x) \in \mathsf{U}$ *and* $\{A_1(t), \ldots, A_n(t)\} \subseteq \mathsf{U}^{k-1}$*. By IH,* $[\{A_1(t), \ldots, A_n(t)\}] \subseteq \mathcal{P}^m$*. Moreover,* $A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow B(x) \in \mathcal{P}$ *by the definition of* U*. Since* $[\phi] = [B(t)] \notin \mathcal{P}^m$*, we have that* $\mathcal{P}^m \neq \mathcal{P}^{\forall}(\mathcal{P}^m)$.

   (b) $A(x) \wedge R(x,y) \rightarrow B(y) \in \mathsf{U}$ *and* $\{A(u), R(u,t)\} \subseteq \mathsf{U}^{k-1}$ *for some term* u*. Analogous to the previous case.*

2. ϕ *is of the form* $R(t,u)$

   (a) $S(x,y) \rightarrow R(x,y) \in \mathsf{U}$ *and* $S(t,u) \in \mathsf{U}^{k-1}$*. Analogous to the previous case.*

   (b) $S(y,z) \rightarrow R(x,y) \in \mathsf{U}$ *and* $S(u,t) \in \mathsf{U}^{k-1}$*. Analogous to the previous case.*

   (c) $S(x,z) \wedge V(z,y) \rightarrow R(x,y) \in \mathsf{U}$ *and* $\{S(t,v), V(v,u)\} \subseteq \mathsf{U}^{k-1}$ *for some term* v*. Analogous to the previous case.*

In every case we have that $\mathcal{P}^m \neq \mathcal{P}^{\forall}(\mathcal{P}^m)$.

Therefore, by Definition 3.5.3, we have that either of the following holds: (x) $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$ *or (xx)* $\mathcal{P}^{m+1} = \mathcal{P}^{\forall}(\mathcal{P}^m)$.

Let (x) $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$*. We first show that* $S(\mathsf{U}^{k-1}, \mathcal{P}^{m+1})$*. By induction hypothesis we have that, for every* $\phi \in \mathsf{U}^{k-1}$*,* $[\phi]^m_{\mathcal{P}} \in \mathcal{P}^m$*. Note that, by Lemma 4.4.4, for every* $\phi \in \mathcal{P}^m$*,* $[\phi]^{m+1}_{\mathcal{P}} \in \mathcal{P}^{m+1}$*. Thus, for every* $\phi \in \mathsf{U}^{k-1}$*,* $[\phi]^{m+1}_{\mathcal{P}} \in \mathcal{P}^{m+1}$*. Furthermore, note that, since* $\mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^m)$*, we can conclude that* $\mathcal{P}^{m+1}$ *contains no successors of* t *(note that by induction hypothesis,* $\mathcal{P}^m$ *contains no such terms). Hence,* $S(\mathsf{U}^{k-1}, \mathcal{P}^{m+1})$.

Let us assume that there is some $\phi \in \mathsf{U}^k$ *such that* $[\phi] \notin \mathcal{P}^{m+1}$ *as otherwise we would have that* $S(\mathsf{U}^k, \mathcal{P}^{m+1})$ *and the induction step would hold. In this case, we may verify* $\mathcal{P}^{m+1} \neq \mathcal{P}^{\forall}(\mathcal{P}^{m+1})$ *making an analogous argument as the one at the beginning of case (i). Thus, the same situation as before arises: We either have that* $\mathcal{P}^{m+2} = \mathcal{R}^{\approx}(\mathcal{P}^{m+1})$ *or* $\mathcal{P}^{m+2} = \mathcal{P}^{\forall}(\mathcal{P}^m)$*. If* $\mathcal{P}^{m+2} = \mathcal{P}^{\approx}(\mathcal{P}^{m+1})$*, we can make an analogous argument to the one at the beginning of case (x) to conclude that* $S(\mathsf{U}^{k-1}, \mathcal{P}^{m+2})$*. Yet again, if* $S(\mathsf{U}^k, \mathcal{P}^{m+2})$ *does not hold, the very same situation could arise: We either have that* $\mathcal{P}^{m+3} = \mathcal{R}^{\approx}(\mathcal{P}^{m+2})$ *or* $\mathcal{P}^{m+3} = \mathcal{P}^{\forall}(\mathcal{P}^m + 2)$.

Eventually, there must be some $\mathcal{P}^n$*, where* $n = m + c$*, such that*

- $\mathcal{P}^{n-1} = \mathcal{R}^{\approx}(\mathcal{P}^{n-2}), \mathcal{P}^{n-2} = \mathcal{R}^{\approx}(\mathcal{P}^{n-3}), \ldots, \mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^{m})$*, and*
- *either* $\mathcal{P}^n = \mathcal{R}^{\forall}(\mathcal{P}^{n-1})$ *or* $S(\mathsf{U}^k, \mathcal{P}^n)$*.*

Note that, for every possible program $\mathcal{P}'$ *and any* $o \geq 0$*, if* $\mathcal{P}^o = \mathcal{P}^{\approx}(\mathcal{P}^{o-1})$*, then set* $\mathcal{P}^o$ *contains at least one term less than* $\mathcal{P}^{\approx}(\mathcal{P}^{o-1})$*. We proceed to show that, if* $\mathcal{P}^n = \mathcal{R}^{\forall}(\mathcal{P}^{n-1})$*, then* $S(\mathsf{U}^k, \mathcal{P}^n)$*.*

It is clear that $\mathcal{P}^n$ *does not contain any successors of* t *since* $\mathcal{P}^n = \mathcal{R}^{\forall}(\mathcal{P}^{n-1})$*,* $\mathcal{P}^{n-1} = \mathcal{R}^{\approx}(\mathcal{P}^{n-2})$*,* $\mathcal{P}^{n-2} = \mathcal{R}^{\approx}(\mathcal{P}^{n-3}), \ldots, \mathcal{P}^{m+1} = \mathcal{P}^{\approx}(\mathcal{P}^{m})$*, and the set* $\mathcal{P}^m$ *does not contain any successors of* t*. Furthermore, note that* $S(\mathsf{U}^{k-1}, \mathcal{P}^{n-1})$*. Thus, it only remains to show that, for every* $\phi \in \mathsf{U}^k$*,* $[\phi] \in \mathcal{P}^n$*.*

A fact $\phi \in \mathsf{U}^k$ *only if one of the following holds:*

1. $\phi \in \mathsf{U}^{k-1}$*. Then,* $[\phi] \in \mathcal{P}^{n-1}$ *since* $S(\mathsf{U}^{k-1}, \mathcal{P}^{n-1})$*, and thus* $[\phi] \in \mathcal{P}^n$*.*
2. $\phi \notin \mathsf{U}^{k-1}$ *and* ϕ *is of the form* $B(t)$*. Then, since* $\phi \in \mathsf{U}^k$ *and* $\phi \notin \mathsf{U}^{k-1}$*, we have that one of the following must hold:*
   (a) $A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow B(x) \in \mathsf{U}$ *and* $\{A_1(t), \ldots, A_n(t)\} \subseteq \mathsf{U}^{k-1}$*. Since* $S(\mathsf{U}^{k-1}, \mathcal{P}^{n-1})$*,* $[\{A_1(t), \ldots, A_n(t)\}] \subseteq \mathcal{P}^{n-1}$*. Moreover,* $A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow B(x) \in \mathcal{P}$ *by the definition of* U*.*
   (b) $A(x) \wedge R(x, y) \rightarrow B(y) \in \mathsf{U}$ *and* $\{A(u), R(u, t)\} \subseteq \mathsf{U}^{k-1}$ *for some term* u*. Analogous to the previous case.*
3. $\phi \notin \mathsf{U}^{k-1}$ *and* ϕ *is of the form* $R(t, u)$
   (a) $S(x, y) \rightarrow R(x, y) \in \mathsf{U}$ *and* $S(t, u) \in \mathsf{U}^{k-1}$*. Analogous to the previous case.*
   (b) $S(y, z) \rightarrow R(x, y) \in \mathsf{U}$ *and* $S(u, t) \in \mathsf{U}^{k-1}$*. Analogous to the previous case.*
   (c) $S(x, z) \wedge V(z, y) \rightarrow R(x, y) \in \mathsf{U}$ *and* $\{S(t, v), V(v, u)\} \subseteq \mathsf{U}^{k-1}$ *for some term* v*. Analogous to the previous case.*

Note that, in either case, $[\phi] \in \mathcal{P}^n$ *and thus,* $S(\mathsf{U}^k, \mathcal{P}^n)$*.*

(xx) Let $\mathcal{P}^{m+1} = \mathcal{P}^{\forall}(\mathcal{P}^m)$*. Then, we may show that* $S(\mathsf{U}^k, \mathcal{P}^{m+1})$ *in an analogous way as we show that* $(\mathsf{U}^k, \mathcal{P}^n))$ *in the second part of case (x).*

Lemma 4.2.6 *Let* $\mathcal{T}$ *be a TBox and* t *a restricted term. Then, for every possible* $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$*,* $t \notin RC(\mathcal{P}(\mathcal{O}))$*.*

Proof 11 *By Definition 4.2.5, we have that* t *is of the form* $f^y_\rho(u)$ *where* $\rho = A(x) \rightarrow \exists y[R(x, y) \wedge B(y)]$*. Note that,* $t \in RC(\mathcal{P}(\mathcal{O}))$ *only if* $u \in RC(\mathcal{P}(\mathcal{O}))$*. Let* i *be the smallest number such that* $u \in \mathcal{P}(\mathcal{O})^i_R$*. Then, by Lemma 4.4.7, there is some* j *such that* $[\mathcal{U}(\mathcal{T}, u)] \subseteq \mathcal{P}(\mathcal{O})^j_R$ *and* $\mathcal{P}(\mathcal{O})^j_R$ *does not contain any successors of* u*. Furthermore, by the definition of a restricted term, we have that there is some* s *with* $\{R([u], s), B(s)\} \subseteq \mathcal{P}(\mathcal{O})^j_R$*. By Lemma 4.4.4, for every* $\mathcal{P}(\mathcal{O})^k_R$ *with* $k \geq j$*, we have that* $[\{R(u, s), B(s)\}] \subseteq \mathcal{P}(\mathcal{O})^k_R$ *and thus, by the definition of the restricted consequences of a TGD, term* t *may not be introduced in any* $\mathcal{P}^k_R$*.*

4.4.3 $\mathcal{V}_\mathcal{T}$ is an Overchase of $\mathcal{T}$

Lemma 4.4.8 *Let $\mathcal{T}$ be some TBox and let $t_1, \ldots, t_n$ be a sequence of terms such that, for every $i = 1, \ldots, n$, $dep(t_1) \leq dep(t_i)$ and $\{\mathsf{Eq}(t_1, t_2), \ldots, \mathsf{Eq}(t_{n-1}, t_n)\} \subseteq \mathcal{V}_\mathcal{T}$. Then, for every $i = 2, \ldots, n$, $\mathsf{Eq}(t_1, t_i) \in \mathcal{V}_\mathcal{T}$.*

Proof 12 *We proof the lemma via induction on the sequence $t_1, \ldots, t_n$. It is clear that the base case holds, since $\mathsf{Eq}(t_1, t_2) \in \mathcal{V}_\mathcal{T}$.*

We proceed to show the induction step; i.e., we verify that $\mathsf{Eq}(t_1, t_{i-1}) \in \mathcal{P}$ implies $\mathsf{Eq}(t_1, t_i) \in \mathcal{P}$ for every $i = 3, \ldots, n$. By definition, we have that $\mathsf{Eq}(t_{i-1}, t_i) \in \mathcal{V}_\mathcal{T}$. Thus, there must be some $A(x) \wedge R(x,y) \wedge B(y) \wedge R(x,z) \wedge B(z) \rightarrow y \approx z \in \mathcal{R}(\mathcal{T})$ and some u such that $\{A(u), R(u, t_{i-1}), B(t_{i-1}), R(u, t_i), B(t_i))\} \subseteq \mathcal{V}_\mathcal{T}$ (note that, a fact of the form $\mathsf{Eq}(s, v)$ may only occur in $\mathcal{V}_\mathcal{T}$ via application of the $\approx$-rule and every EGD in $\mathcal{R}(\mathcal{T})$ is of the form $C(x) \wedge S(x,y) \wedge D(y) \wedge S(x,z) \wedge D(z) \rightarrow y \approx z$). By IH, we have that $\mathsf{Eq}(t_1, t_{i-1})$ and $dep(t_1) \leq dep(t_{i-1})$. Hence, $\{R(u, t_1), B(t_1)\} \subseteq \mathcal{V}_\mathcal{T}$ by application of the $\exists$-rule. Thus, $\mathsf{Eq}(t_1, t_i) \in \mathcal{V}_\mathcal{T}$ by application of the $\approx$-rule.

Lemma 4.2.9 *Set $\mathcal{V}_\mathcal{T}$ is an overchase of the TBox $\mathcal{T}$.*

Proof 13 *To show the lemma, we verify that, for every ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and every $i \geq 0$, the following hold.*

1. *If $B(t) \in \mathcal{P}(\mathcal{O})^i_R$ where $B \in N_C$, then $B(t)_* \in \mathcal{V}_\mathcal{T}$.*
2. *If $R(t,u) \in \mathcal{P}(\mathcal{O})^i_R$ where $R \in N_R$, then $R(t,u)_* \in \mathcal{V}_\mathcal{T}$.*

We proceed to show (a) and (b) via induction on the chase sequence of $\mathcal{P}(\mathcal{O})$. Our argument is structured as follows.

- *Base Case: Claims (a) and (b) hold for $\mathcal{P}(\mathcal{O})^1_R$.*
- *Induction Step: For every $i > 1$, claims (a) and (b) hold for $\mathcal{P}(\mathcal{O})^i_R$ provided such claims hold for $\mathcal{P}(\mathcal{O})^{i-1}_R$*

*(Base Case) The base case holds since $\mathcal{P}(\mathcal{O})^0_R = \mathcal{A}$ and $\mathcal{I}_\star(\mathcal{T}) \subseteq \mathcal{V}_\mathcal{T}$. Note that, by assumption, there are neither concept nor role names in $\mathcal{A}$ which do not occur in $\mathcal{T}$ (see Section **??**).*

(Induction Step) Let $\mathcal{P}(\mathcal{O}) = \langle \mathcal{R}, \mathcal{I} \rangle$. By the definition of the chase sequence, three possible cases arise: For every $i \geq 1$, $\mathcal{P}(\mathcal{O})^i_R$ is either (i) $\mathcal{R}^\exists_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$, (iii) $\mathcal{R}^\forall_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$ or (iii) $\mathcal{R}^\approx(\mathcal{P}(\mathcal{O})^{i-1}_R)$. We proceed to show that, in either case, the induction hypothesis (IH) holds.

Let (i) $\mathcal{P}(\mathcal{O})^i_O = \mathcal{R}^\exists_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$. We do a case by cases analysis of all the possible facts that may occur in $\mathcal{P}(\mathcal{O})^i_O$ and verify that the induction step holds. When checking if the induction step holds for some given fact $\phi \in \mathcal{P}(\mathcal{O})^i_O$ we assume that $\phi \notin \mathcal{P}(\mathcal{O})^{i-1}_O$ as otherwise the induction step trivially holds by induction hypothesis (IH).

- *$B(t) \in \mathcal{P}(\mathcal{O})^i_R$ only if $\rho = A(x) \rightarrow \exists y[R(x,y) \wedge B(y)] \in \mathcal{R}(\mathcal{T})$ and $t = f^y_\rho(u)$. By Lemma 4.2.6, t is not restricted since t occurs in the restricted chase of $\mathcal{P}(\mathcal{O})$. Then, by Definition 4.2.5, t_* is*

not restricted as well. By IH, $A(u)_* \in \mathcal{V}_\mathcal{T}$. *Thus, by application of the* $\exists$*-rule from Figure 4.1,* $B(t)_* \in \mathcal{V}_\mathcal{T}$.

- $R(t,u) \in \mathcal{P}(\mathcal{O})^i_R$ *only if* $\rho = A(x) \to \exists y[R(x,y) \wedge B(y)] \in \mathcal{P}(\mathcal{O})$ *and* $u = f^y_\rho(t)$. *Analogous to the previous case.*

Thus, the induction step holds if $\mathcal{P}(\mathcal{O})^i_O = \mathcal{R}^\exists_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$.

Let (ii) $\mathcal{P}(\mathcal{O})^i_R = \mathcal{R}^\forall_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$. *Again, we do a case by cases analysis of all the possible facts that may occur in* $\mathcal{P}(\mathcal{O})^i_O$ *and verify that claims (a) and (b) hold.*

- $B(t) \in \mathcal{P}(\mathcal{O})^i_R$ *only if one of the following holds.*
  - $A_1(x) \wedge \ldots \wedge A_n(x) \to B(x) \in \mathcal{R}(\mathcal{T})$ *and* $\{A_1(t), \ldots, A_n(t)\} \subseteq \mathcal{P}(\mathcal{O})^{i-1}_R$. *By IH,* $\{A_1(t), \ldots, A_n(t)\}_* \subseteq \mathcal{V}_\mathcal{T}$. *Thus, by application of the* $\forall$*-rule from Figure 4.1,* $B(t)_* \in \mathcal{V}_\mathcal{T}$.
  - $A(x) \wedge R(x,y) \to B(y) \in \mathcal{P}(\mathcal{O})$ *and* $\{A(u), R(u,t)\} \subseteq \mathcal{P}(\mathcal{O})^{i-1}_O$. *Analogous to the previous case.*
- $R(t,u) \in \mathcal{P}(\mathcal{O})^i_R$ *only if one of the following holds.*
  - $S(x,y) \to R(x,y) \in \mathcal{P}(\mathcal{O})$ *and* $S(t,u) \in \mathcal{P}(\mathcal{O})^{i-1}_R$. *Analogous to case the previous case.*
  - $S(y,x) \to R(x,y) \in \mathcal{P}(\mathcal{O})$ *and* $S(u,t) \in \mathcal{P}(\mathcal{O})^{i-1}_R$. *Analogous to the previous case.*
  - $S(x,z) \wedge V(z,y) \to R(x,y) \in \mathcal{P}(\mathcal{O})$ *and* $\{S(u,v), V(v,t)\} \subseteq \mathcal{P}(\mathcal{O})^{i-1}_R$. *Analogous to the previous case.*

Thus, the induction step holds if $\mathcal{P}(\mathcal{O})^i_O = \mathcal{R}^\forall_R(\mathcal{P}(\mathcal{O})^{i-1}_R)$.

Let (iii) $\mathcal{P}(\mathcal{O})^i_R = \mathcal{R}^{\approx}\mathcal{P}(\mathcal{O})^{i-1}_R$. *Again, we do a case by cases analysis of all the possible facts that may occur in* $\mathcal{P}(\mathcal{O})^i_O$ *and verify that both (a) and (b) hold.*

- $B(t) \in \mathcal{P}(\mathcal{O})^i_R$ *only if there exists a sequence of terms* $t_1, \ldots, t_n$ *such that* $t_1 = t$; $B(t_n) \in \mathcal{P}(\mathcal{O})^{i-1}_R$; *for every* $j = 1, \ldots, n$, $dep(t_1) \leq dep(t_j)$; *and, for every* $j = 1, \ldots, n-1$, *there is some EGD* $A_j(x) \wedge R_j(x,y) \wedge C_j(y) \wedge R_j(x,y) \wedge C_j(y) \in \mathcal{R}(\mathcal{T})$ *and some* u_j *with* $\{A_j(u_j), R_j(u_j,t_j), C_j(t_j), R_j(u_j,t_{j+1}), C_j(t_{j+1})\} \subseteq \mathcal{P}(\mathcal{O})^i_R$. *By IH, for every* $j = 1, \ldots, n-1$, $\{A_j(u_j), R_j(u_j,t_j), C_j(t_j), R_j(u_j,t_{j+1}), C_j(t_{j+1})\}_* \subseteq \mathcal{V}_\mathcal{T}$. *Thus,* $\{\mathsf{Eq}(t_1,t_2), \ldots, \mathsf{Eq}(t_{n-1},t_n)\}_* \subseteq \mathcal{V}_\mathcal{T}$ *by application of the Eq-rule. By Lemma 4.4.8,* $\mathsf{Eq}(t_1,t_n)_* \in \mathcal{V}_\mathcal{T}$ *(note that the lemma is indeed applicable since, for every* $j = 1, \ldots, n$, $dep(t^1_*) \leq dep(t^n_*)$*). Thus,* $B(t)_* \in \mathcal{V}_\mathcal{T}$ *by application of the* $\forall$*-rule.*

- $R(t,u) \in \mathcal{P}(\mathcal{O})^i_R$ *(resp.,* $R(u,t) \in \mathcal{P}(\mathcal{O})^i_R$*) only if there exists a sequence of terms* $t_1, \ldots, t_n$ *such that* $t_1 = t$; $R(t_n,u) \in \mathcal{P}(\mathcal{O})^{i-1}_R$ *(resp.,* $R(t_n,u) \in \mathcal{P}(\mathcal{O})^i_R$*); for every* $j = 1, \ldots, n$, $dep(t_1) \leq dep(t_j)$; *and, for every* $j = 1, \ldots, n-1$, *there is some EGD* $A_j(x) \wedge R_j(x,y) \wedge C_j(y) \wedge R_j(x,y) \wedge C_j(y) \in \mathcal{R}(\mathcal{T})$ *and some* u_j *with* $\{A_j(u_j), R_j(u_j,t_j), C_j(t_j), R_j(u_j,t_{j+1}), C_j(t_{j+1})\} \subseteq \mathcal{P}(\mathcal{O})^i_R$. *Analogous to the previous case.*

Thus, the induction step holds if $\mathcal{P}(\mathcal{O})^i_O = \mathcal{R}^{\approx}\mathcal{P}(\mathcal{O})^{i-1}_R$.

4.4.4 Complexity Results

Lemma 4.4.9 *Let $\mathcal{T}$ be some TBox and let t be some term which may occur during the computation of the chase of a program $\mathcal{P}(\langle \mathcal{T}, \mathcal{A} \rangle)$. If t is not n-cyclic, then deciding whether t is restricted is in* PTime *with respect to $\mathcal{T}$.*

Proof 14 *We can decide whether t is restricted by computing the restricted chase of the program $\mathcal{U}(\mathcal{T}, t)$. If t is non-n-cyclic, then $dep(t) \leq n|\mathcal{T}^{\exists}|$: Note that (i) t is of the form $f_1(\dots f_n(*))$, the number of function symbols in $\mathcal{V}_{\mathcal{T}}$ is at most $|\mathcal{T}^{\exists}|$, and a term $f_1(\dots f_n(*))$ is n-cyclic if and only if it contains $n+1$ occurrences of the same function symbol. Thus, by definition, $|\mathcal{I}(t)|$ is linear with respect to $|\mathcal{T}^{\exists}|$ and so is the number of terms in it.*

It is clear that the restricted chase of $\mathcal{U}(\mathcal{T}, t)$ can be computed in polynomial time. Note that: (i) $\mathcal{U}(\mathcal{T}, t)$ does not contain any TGDs featuring existentially quantified variables, (ii) every rule in $\mathcal{U}(\mathcal{T}, t)$ contains at most 3 different variables, (iii) every predicate in $\mathcal{U}(\mathcal{T}, t)$ has a maximum arity of 2 and (iv) once a functional term is removed from the restricted chase sequence due to some replacement, it is never reintroduced again (see Lemma 4.4.1).

Lemma 4.2.14 *Deciding whether some TBox $\mathcal{T}$ is RCA_n is in* ExpTime.

Proof 15 *By the definition of RCA_n: $\mathcal{T}$ is RCA_n if and only if there is some n-cyclic term in $\mathcal{V}_{\mathcal{T}}$. Thus, it suffices to compute the set $\mathcal{V}_{\mathcal{T}}$ up to the occurrence of the first n-cyclic term. Hence, for the rest of the argument, we assume that $\mathcal{V}_{\mathcal{T}}$ may not include n-cyclic terms.*

Every term t in $\mathcal{V}_{\mathcal{T}}$ has a depth of at most $n|\mathcal{T}^{\exists}|$: Note that (i) t is of the form $f_1(\dots f_n(*))$, the number of function symbols in $\mathcal{V}_{\mathcal{T}}$ is at most $|\mathcal{T}^{\exists}|$, and a term $f_1(\dots f_n(*))$ is n-cyclic if and only if it contains $n+1$ occurrences of the same function symbol. Thus, there exist at most $\mathsf{T} = \sum_{i=0}^{n|\mathcal{T}^{\exists}|} |\mathcal{T}^{\exists}|^i$ non-n-cyclic terms that may occur in $\mathcal{V}_{\mathcal{T}}$.

Every axiom in $\mathcal{T}$ may contain at most 3 different concept or role names and there are at most $|N_C(\mathcal{T}) + N_R(\mathcal{T})|$ different predicates in $\mathcal{V}_{\mathcal{T}}$ where $N_C(\mathcal{T})$ and $N_R(\mathcal{T})$ are the sets of concept names and role names in $\mathcal{T}$. Thus, the number of predicates in $\mathcal{V}_{\mathcal{T}}$ is at most $\mathsf{P} = 3|\mathcal{T}|$. Thus, the maximum number of facts that may occur in $\mathcal{V}_{\mathcal{T}}$ is $\mathsf{F} = \mathsf{P}\mathsf{T}^2$ since the arity of every predicate is at most two. Hence, since all of the rules from Figure 4.1 result in the addition of at least some fact to $\mathcal{V}_{\mathcal{T}}$, it takes at most T many applications of such rules to compute $\mathcal{V}_{\mathcal{T}}$. It can be verified that T is exponentially large with respect to $n|\mathcal{T}|$.

Finally, note that all of the expansion rules can be applied in polynomial time. This is also the case for the $\exists$-rule: Given some term non-cyclic t, we can decide whether t is restricted in polynomial time (see Lemma 4.4.9).

Lemma 4.4.10 *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be some RCA_n ontology and γ a query. Then, checking whether $\mathcal{O} \models \gamma$ is in* ExpTime.

Proof 16 *As shown in Section 3.5, it suffices to compute the restricted chase of $\mathcal{P}(\mathcal{O})$ to decide whether $\mathcal{O} \models \gamma$. We proceed to show that $RC(\mathcal{P}(\mathcal{O}))$ can be computed in* ExpTime.

As shown in the proof of Lemma 4.2.14, if $\mathcal{T}$ is RCA_n then $\mathcal{V}_\mathcal{T}$ contains at most F facts, a number which is exponential with respect to $n|\mathcal{T}^\exists|$ (where $\mathcal{T}^\exists$ is the set of all existential axioms in $\mathcal{T}$). By the definition of an overchase we have that $\phi \in RC(\mathcal{P}(\mathcal{O}))$ only if $\phi_* \in \mathcal{V}_\mathcal{T}$. Thus, the maximum number of facts in $RC(\mathcal{P}(\langle\mathcal{T}, \mathcal{A}\rangle))$ is $\mathsf{T}|N_I(\mathcal{O})|^2$ where $N_I(\mathcal{O})$ is the number of individuals in the ontology $\mathcal{O}$. Note that every predicate in $\mathcal{P}(\langle\mathcal{T}, \mathcal{A}\rangle)$ has a maximum arity of 2.

Furthermore, even if some facts are removed from the restricted chase sequence during its computation, this is not problematic: See Lemma 4.4.2 to verify that, once a fact is removed from the restricted chase sequence, it may not be reintroduced again.

Lemma 4.4.11 *Let $\mathcal{O} = \langle\mathcal{T}, \mathcal{A}\rangle$ be some RCA_n ontology and γ a query. Then, checking whether $\mathcal{O} \models \gamma$ is in* ExpTime*-hard.*

Proof 17 *It is shown in [Cuenca Grau et al. 2013] that weak acyclicity is less general than universal MFA when it comes to Horn-$\mathcal{SRI}$ TBoxes. It can be easily verified that, when it comes to Horn-$\mathcal{SRI}$ TBoxes, universal MFA and RCA_1 coincide in terms of generality. I.e., a Horn-$\mathcal{SRI}$ TBox is universally MFA if and only if it is RCA_1. Thus, the Lemma follows from Lemma 59 from [Cuenca Grau et al. 2013] which states the following: "Let $\mathcal{T}$ be a weak acyclic Horn-$\mathcal{SRI}$ TBox, let $\mathcal{I}$ be an instance and let ϕ be a fact. Then, checking whether $\langle\mathcal{R}(\mathcal{T}), \mathcal{I}\rangle \models \phi$ is* ExpTime*-hard."*

Lemma 4.2.15 *Let $\mathcal{O} = \langle\mathcal{T}, \mathcal{A}\rangle$ be some RCA_n ontology and γ a query. Then, checking whether $\mathcal{O} \models \gamma$ is* ExpTime*-complete with respect to $|\mathcal{T}|$ and n.*

Proof 18 *Follows from Lemmas 4.4.10 and 4.4.11.*

4.4.5 $\mathbf{RCA}_n$ vs $\mathbf{MFA}^\cup$

In an attempt to make this document more self-contained, we present a summarized definition of $\text{MFA}^\cup$ that can be directly applied to Horn-$\mathcal{SRIQ}$ TBoxes.

Definition 4.4.12 *Let $\mathsf{Eq} = \{\top(x) \to \mathsf{Eq}(x, x), \mathsf{Eq}(y, x) \to \mathsf{Eq}(x, y), \mathsf{Eq}(x, z) \wedge \mathsf{Eq}(z, y) \to \mathsf{Eq}(x, y)\}$ and let Ξ be the function mapping axioms to rules which is identical to π from Figure 3.2 in all but the following cases:*

Given some TBox $\mathcal{T}$, let $MFA^\cup(\mathcal{T}) = \langle\pi(\mathcal{T}) \cup \mathsf{Eq}, \mathcal{I}_\star(\mathcal{T})\rangle$. A TBox $\mathcal{T}$ is $MFA^\cup$ if and only if there are no cyclic terms in the oblivious chase of $MFA^\cup(\mathcal{T})$.

It can be easily verified that our definition coincides with the on from [Cuenca Grau et al. 2013]. We proceed a theoretical comparison of $\text{MFA}^\cup$ and RCA_n.

Lemma 4.2.17 *If $\mathcal{T}$ is $MFA^\cup$ then $\mathcal{T}$ is RCA_n for every $n > |\mathcal{T}_\exists|$.*

Proof 19 *Let $\mathcal{T}$ be some $MFA^\cup$ TBox. We proceed to show that $\mathcal{T}$ is RCA_n for every $n > |\mathcal{T}_\exists|$.*

By the definition of $MFA^\cup$, we have that the set $OC(MFA^\cup(\mathcal{T}))$ does not contain any cyclic terms. Thus, for every term $OC(MFA^\cup(\mathcal{T}))$, $dep(t) \leq |\mathcal{T}^\exists|$ since, by definition, a term s with $dep(s) > |\mathcal{T}^\exists|$ is necessarily cyclic.

We proceed to show that this upper bound on the depth of terms also holds for $\mathcal{V}_\mathcal{T}$. Note that, if such is the case, then $\mathcal{T}$ is necessarily RCA_n for every $n > |\mathcal{T}^\exists|$. This is because $\mathcal{T}$ is RCA_n if and only if some n-cyclic term occurs in $\mathcal{V}_\mathcal{T}$, and, for every n-cyclic term u, $dep(u) > |\mathcal{T}^\exists|$. Thus, the lemma follows if we verify that, for every term $t \in \mathcal{V}_\mathcal{T}$, $dep(t) \leq |\mathcal{T}^\exists|$.

We show that $|\mathcal{T}^\exists|$ is an upper bound on the depth on the terms in $\mathcal{V}_\mathcal{T}$ by constructing a set $\mathcal{W}_\mathcal{T}$ with the following properties:

1. *If there is some term $t \in \mathcal{W}_\mathcal{T}$, then there is some term $u \in MFA^\cup(\mathcal{T})$ with $dep(t) \leq dep(u)$.*
2. *A term t occurs in $\mathcal{V}_\mathcal{T}$ only if it occurs in $\mathcal{W}_\mathcal{T}$.*

We proceed with the construction of $\mathcal{W}_\mathcal{T}$. We first present a function, which will be used in the definition of such set. Let Ψ be the function mapping TBox axioms into rules defined as follows: Where $\rho = A(x) \rightarrow \exists y R(x,y) \wedge B(y)$.

Let $\mathcal{W}^1_\mathcal{T}, \mathcal{W}^2_\mathcal{T}, \mathcal{W}^3_\mathcal{T}, \ldots$ be the sequence such that $\mathcal{W}^1_\mathcal{T} = \{A(\star,\star) \mid A \in N_C(\mathcal{T})\} \cup \{R(\star,\star,\star,\star) \mid R \in N_R(\mathcal{T})\}$ and, for every $i \geq 2$, $\mathcal{W}^i_\mathcal{T}$ is the set that results by applying some rule in Figure 4.3 to $\mathcal{W}^{i-1}_\mathcal{T}$. Furthermore, let $\mathcal{W}_\mathcal{T} = \bigcup_{i \in \mathbb{N}} \mathcal{W}^i_\mathcal{T}$.

We proceed to show some properties of $\mathcal{W}_\mathcal{T}$, which will be used in subsequent parts of this argument. For all terms t t', u and u', and predicates A and R, the following hold.

1. *If $A(t,t')$, $R(t,t',u,u')$ or $R(u,u',t,t')$ in $\mathcal{W}_\mathcal{T}$, then $dep(t) \leq dep(t')$.*
2. *If $A(t,t') \in \mathcal{W}_\mathcal{T}$, then $A(t',t') \in \mathcal{W}_\mathcal{T}$.*
3. *If $R(t,t',u,u') \in \mathcal{W}_\mathcal{T}$, then $R(t',t',u,u') \in \mathcal{W}_\mathcal{T}$.*
4. *If $R(t,t',u,u') \in \mathcal{W}_\mathcal{T}$, then $R(t,t',u',u') \in \mathcal{W}_\mathcal{T}$.*

These claims can be proven via induction via induction on the sequence $\mathcal{W}^1_\mathcal{T}$, $\mathcal{W}^2_\mathcal{T}$, $\mathcal{W}^3_\mathcal{T}, \ldots$. It is clear that the base case holds since $\mathcal{W}^1_\mathcal{T} = \{A(\star,\star) \mid A \in N_C(\mathcal{T})\} \cup \{R(\star,\star,\star,\star) \mid R \in N_R(\mathcal{T})\}$. The induction step can be shown with a simple case by case analysis on all the possible facts that may be derived during the computation of the sequence $\mathcal{W}^1_\mathcal{T}$, $\mathcal{W}^2_\mathcal{T}$, $\mathcal{W}^3_\mathcal{T}, \ldots$.

4.4.5.1 Claim (a):

We proceed to show the following claim: If there is some term $t \in \mathcal{W}_\mathcal{T}$, then there is some term $u \in MFA^\cup(\mathcal{T})$ with $dep(t) \leq dep(u)$.

To do so, we first introduce a function π which maps the terms in $\mathcal{W}_\mathcal{T}$ into terms in $MFA^\cup(\mathcal{T})$. Let π be some function such that $\pi(\star) = \star$ and, for every term of the form $t = f^y_\rho(u)$ where $\rho = A(x) \rightarrow \exists y[R(x,y) \wedge B(y)]$, $\pi(t) = f^y_\rho(\pi(u'))$ where u' is some term such that $A(u,u') \in \mathcal{W}_\mathcal{T}$. By claim (ii), term t may not occur in $\mathcal{W}_\mathcal{T}$ unless there is some fact of the form $A(u,u') \in \mathcal{W}_\mathcal{T}$ and thus, π is well-defined. Furthermore, since $dep(u) \leq dep(u')$ for every $A(u,u') \in \mathcal{W}_\mathcal{T}$ by claim (i), we have that $dep(t) \leq dep(\pi(t))$ for every term t.

We proceed to show the following claims via induction.

1. *If t occurs in $\mathcal{W}_\mathcal{T}$, then $\pi(t)$ occurs in $OC(MFA^\cup(\mathcal{T}))$.*

2. $A(t,t') \in \mathcal{W}_\mathcal{T}$ *implies* $\{\mathsf{Eq}(\pi(t),\pi(t')), A(\pi(t'))\} \subseteq OC(MFA^\cup(\mathcal{T}))$.

3. $R(t,t',u,u') \in \mathcal{W}_\mathcal{T}$ *implies* $\{\mathsf{Eq}(\pi(t),\pi(t')), R(\pi(t'),\pi(u')), \mathsf{Eq}(\pi(u'),\pi(u))\} \subseteq OC(MFA^\cup(\mathcal{T}))$.

4. $\mathsf{Eq}(t,u) \in \mathcal{W}_\mathcal{T}$ *implies* $\mathsf{Eq}(\pi(t),\pi(u)) \in OC(MFA^\cup\mathcal{T})$.

Note that, claim (a) directly follows from (1) and the fact that $dep(t) \leq dep(\pi(t))$ *for every term* $\mathcal{T}$ *in* $\mathcal{W}_\mathcal{T}$. *The remainder of the claims (2-4) are just introduced to properly structure our induction argument.*

The base case of the induction trivially holds since $\mathcal{I}_\star(\mathcal{T}) \subseteq OC(MFA^\cup(\mathcal{T}))$ *and* $\mathcal{W}^1_\mathcal{T} = \{A(\star,\star) \mid A \in N_C(\mathcal{T})\} \cup \{R(\star,\star,\star,\star) \mid R \in N_R(\mathcal{T})\}$. *We proceed to show the induction step; i.e., we show that (1-4) hold for every fact* $\phi \in \mathcal{W}^i_\mathcal{T}$ *provided they hold for every fact in* $\mathcal{W}^i_\mathcal{T}$. *When showing that the claims hold for some term or fact* $\phi \in \mathcal{W}^i_\mathcal{T}$ *we assume that* $\phi \notin \mathcal{W}^{i-1}_\mathcal{T}$ *as otherwise the induction step trivially holds by induction hypothesis.*

Let t *be some term occurring in* $\mathcal{W}^i_\mathcal{T}$. *Then,* t *is of the form* $t = f^y_\rho(u)$ *for some term* u *where* $\rho = A(x) \to \exists y[R(x,y) \wedge B(y)]$, $A(x,x') \to R(x,x',f^y_\rho(x),f^y_\rho(x)) \wedge B(f^y_\rho(x),f^y_\rho(x)) \in \Psi(\mathcal{T})$ *and* $A(u,u') \in \mathcal{W}^{i-1}_\mathcal{T}$. *By the definition of* π, *we have that* $t = f^y_\rho(u) = f^y_\rho(s')$ *where* s' *is some term such that* $A(u,s') \in \mathcal{W}^{i-1}_\mathcal{T}$. *By IH,* $A(s') \in OC(MFA^\cup(\mathcal{T}))$. *It is clear that claim (1) holds since* $\rho \in MFA^\cup(\mathcal{T})$.

Let ϕ *be a fact of the form* $A(t,t') \in \mathcal{W}^i_\mathcal{T}$. *Then, one of the following cases must hold.*

- $\bigwedge^n_{i=1} A(x,x'_i) \to \bigwedge^n_{i=1}(B(x,x'_i) \wedge B(x'_i,x'_i)) \in \Psi(\mathcal{T})$ *and* $\{A_i(t,t'_i) \mid i = 1,\ldots,n\}$. *Then,* $\bigwedge^n_{i=1} A_i(x_i) \wedge \bigwedge^{n-1}_{i=1} \mathsf{Eq}(x_i,x_{i+1}) \to \bigwedge^n_{i=1} B(x_i) \in MFA^\cup(\mathcal{T})$. *By IH,* $\{\mathsf{Eq}(\pi(t),\pi(t'_i)), A_i(\pi(t'_i)) \mid i = 1,\ldots,n\}$.

- $A(x,x') \wedge R(x,x'',y,y') \to B(y,y') \in \Psi(\mathcal{T})$ *and* $\{A(u,u'), R(u,u'',t,t')\} \subseteq \mathcal{W}^{i-1}_\mathcal{T}$. *Then* $A(x) \wedge \mathsf{Eq}(x,x') \wedge R(x',y) \to B(y) \in MFA^\cup(\mathcal{T})$. *By IH,* $\{\mathsf{Eq}(\pi(u),\pi(u'))$, $A(\pi(u')), \mathsf{Eq}(\pi(u),\pi(u''))$, $R(\pi(u''),\pi(t')), \mathsf{Eq}(\pi(t),\pi(t'))\} \subseteq OC(MFA^\cup(\mathcal{T}))$.

- $A(x,x') \to R(x,x',f^y_\rho(x),f^y_\rho(x)) \wedge B(f^y_\rho(x),f^y_\rho(x)) \in \Psi(\mathcal{T})$ *where* $\rho = A(x) \to \exists y[R(x,y) \wedge B(y)]$, $A(u,u') \in \mathcal{W}^{i-1}_\mathcal{T}$, $t = t' = f^y_\rho(u)$. *By the definition of* π, *we have that* $\pi(t) = f^y_\rho(s')$ *where* s' *is some term such that* $A(u,s') \in \mathcal{W}^{i-1}_\mathcal{T}$. *By IH,* $A(\pi(s')) \in OC(MFA^\cup(\mathcal{T}))$. *Also, note that* $A(x) \to \exists y[R(x,y) \wedge B(y)] \in MFA^\cup(\mathcal{T})$.

- $\{\mathsf{Eq}(t,s), A(s,t')\} \subseteq \mathcal{W}^{i-1}_\mathcal{T}$ *and* $dep(t) \leq dep(s)$. *By IH,* $\{\mathsf{Eq}(\pi(t),\pi(s))$, $\mathsf{Eq}(\pi(s),\pi(t'))$, $A(\pi(t'))\}$ $\subseteq OC(MFA^\cup(\mathcal{T}))$.

It is clear that in either case, claim (2) holds. Note that $\{\top(x) \to \mathsf{Eq}(x,x), \mathsf{Eq}(y,x) \to \mathsf{Eq}(x,y)$, $\mathsf{Eq}(x,z) \wedge \mathsf{Eq}(z,y) \to \mathsf{Eq}(x,y)\} \subseteq MFA^\cup(\mathcal{T})$.

Let $\phi \in \mathcal{W}_i$ *be a fact of the form* $S(t,t',u,u')$. *Then,* $S(t,t',u,u') \in \mathcal{W}^i_\mathcal{T}$ *only if one of the following cases hold.*

- $A(x,x') \to S(x,x',f^y_\rho(x),f^y_\rho(x)) \wedge B(f^y_\rho(x),f^y_\rho(x)) \in \Psi(\mathcal{T})$ *and* $A(t,t') \in \mathcal{W}^{i-1}_\mathcal{T}$. *Note that, in this case,* $f^y_\rho(t) = u = u'$. *By the definition of* π, *we have that* $\pi(f^y_\rho(t)) = \pi(u) = f^y_\rho(s')$

where s' is some term such that $A(t,s') \in \mathcal{W}_{\mathcal{T}}^{i-1}$. By IH, $A(\pi(s')) \in OC(MFA^{\cup}(\mathcal{T}))$. Also, $A(x) \rightarrow \exists y[R(x,y) \wedge B(y)] \in MFA^{\cup}(\mathcal{T})$.

- *$R(x,x',y,y') \rightarrow S(x,x',y,y') \in \Psi(\mathcal{T})$ and $R(t,t',u,u') \in \mathcal{W}_{\mathcal{T}}^{i-1}$. Then, $R(x,y) \rightarrow S(x,y) \in MFA^{\cup}(\mathcal{T})$. By IH, $\{\mathsf{Eq}(\pi(t),\pi(t')), R(\pi(t'),\pi(u')), \mathsf{Eq}(\pi(u'),\pi(u))\} \subseteq OC(MFA^{\cup}(\mathcal{T}))$.*

- *$R(y,y',x,x') \rightarrow S(x,x',y,y') \in \Psi(\mathcal{T})$ and $R(u,u',t,t') \in \mathcal{W}_{\mathcal{T}}^{i-1}$. Then, $R(y,x) \rightarrow S(x,y) \in MFA^{\cup}(\mathcal{T})$. By IH, $\{\mathsf{Eq}(\pi(u),\pi(u'))$, $R(\pi(u'),\pi(t'))$, $\mathsf{Eq}(\pi(t'),\ \pi(t))\} \subseteq OC(MFA^{\cup}(\mathcal{T}))$.*

- *$R(y,y',z,z) \wedge V(z,z'',y,y') \rightarrow S(x,x',y,y') \in \Psi(\mathcal{T})$ and $\{R(t,t',s,s')$, $S(s,\ s'',\ u,u')\} \subseteq \mathcal{W}_{\mathcal{T}}^{i-1}$. Then, $R(x,z) \wedge \mathsf{Eq}(z,z') \wedge R(z',y) \rightarrow S(x,y) \in MFA^{\cup}(\mathcal{T})$. By IH, $\{\mathsf{Eq}(\pi(t),\pi(t')), R(\pi(u'),\pi(t'))$, $\mathsf{Eq}(\pi(s'),\pi(s)), \mathsf{Eq}(\pi(s),\pi(s''))$, $R(\pi(s'''),\pi(u'))$, $\mathsf{Eq}(\pi(u'),\pi(u))\} \subseteq \mathcal{W}_{\mathcal{T}}^{i-1}$.*

It is clear that in either case, claim (3) holds. Again, note that note that $\{\top(x) \rightarrow \mathsf{Eq}(x,x), \mathsf{Eq}(y,x) \rightarrow \mathsf{Eq}(x,y)$, $\mathsf{Eq}(x,z) \wedge \mathsf{Eq}(z,y) \rightarrow \mathsf{Eq}(x,y)\} \subseteq MFA^{\cup}(\mathcal{T})$.

Let ϕ be a fact of the form $\mathsf{Eq}(t,u)$. Then, $\mathsf{Eq}(t,u) \in \mathcal{W}_{\mathcal{T}}^{i}$ only if one of the following cases hold.

- *$A(x,x') \wedge R(x,x'',y,y') \wedge B(y,y'') \wedge R(x,x''',z,z') \wedge B(z,z'') \rightarrow \mathsf{Eq}(y,z) \in \mathcal{W}_{\mathcal{T}}$ and $\{A(s,s')$, $R(s,s'',t,t'), B(t,t''), R(s,s''',u,u'), B(u,u'')\} \subseteq \mathcal{W}_{\mathcal{T}}^{i-1}$. By IH, $\{\mathsf{Eq}(\pi(s),\pi(s')), A(\pi(s'))$, Eq $(\pi(s),\pi(s')), R(\pi(s'),\pi(t'))$, $\mathsf{Eq}(\pi(t'),\pi(t)), \mathsf{Eq}(\pi(t),\pi(t'')), B(\pi(t''))$, $\mathsf{Eq}(\pi(s),\pi(s''')), R(\pi(s'''),$ $\pi(u'))$, $\mathsf{Eq}(\pi(u'),\pi(u))$, $\mathsf{Eq}(\pi(u),\pi(u''))$, $B(\pi(u''))\} \subseteq OC(MFA^{\cup}(\mathcal{T}))$. Also, $A(x) \wedge \mathsf{Eq}(x,x') \wedge R(x',y) \wedge \mathsf{Eq}(y,y') \wedge \mathsf{Eq}(x,x'') \wedge R(x'',z) \wedge \mathsf{Eq}(z,z') \wedge B(z') \rightarrow \mathsf{Eq}(y,z) \in MFA^{\cup}(\mathcal{T})$.*

- *$\mathsf{Eq}(u,t) \in \mathcal{W}_{\mathcal{T}}^{i-1}$. By IH, $\mathsf{Eq}(\pi(u),\pi(t)) \in OC(MFA^{\cup}(\mathcal{T}))$.*

- *$\{\mathsf{Eq}(t,s), \mathsf{Eq}(s,u)\} \subseteq \mathcal{W}_{\mathcal{T}}^{i-1}$. By IH, $\{\mathsf{Eq}(t,s), \mathsf{Eq}(s,u)\} \subseteq OC(MFA^{\cup}(\mathcal{T}))$.*

It is clear that in either case, claim (3) holds. Again, note that note that $\{\top(x) \rightarrow \mathsf{Eq}(x,x), \mathsf{Eq}(y,x) \rightarrow \mathsf{Eq}(x,y)$, $\mathsf{Eq}(x,z) \wedge \mathsf{Eq}(z,y) \rightarrow \mathsf{Eq}(x,y)\} \subseteq MFA^{\cup}(\mathcal{T})$.

4.4.5.2 Claim (b): A term t occurs in $\mathcal{V}_{\mathcal{T}}$ only if it occurs in $\mathcal{W}_{\mathcal{T}}$.

Claim (a) follows from the following statements.

1. *For every term t, if $t \in \mathcal{V}_{\mathcal{T}}$, then $t \in \mathcal{W}_{\mathcal{T}}$.*
2. *For every term t and every $A \in N_C$, if $A(t) \in \mathcal{V}_{\mathcal{T}}$, then $A(t,t) \in \mathcal{W}_{\mathcal{T}}$*
3. *For every terms pair of terms t and u, and every role name R, if $R(t,u) \in \mathcal{V}_{\mathcal{T}}$, then $R(t,t,u,u) \in \mathcal{W}_{\mathcal{T}}$.*
4. *For all terms t and u, if $\mathsf{Eq}(t,u) \in \mathcal{V}_{\mathcal{T}}$, then $\mathsf{Eq}(t,u) \in \mathcal{W}_{\mathcal{T}}$.*

Claim (b) follows from (1): The rest of the claims (2-4) are in placed to properly structure the induction argument.

Let $\mathcal{V}_{\mathcal{T}}^1$, $\mathcal{V}_{\mathcal{T}}^2, \ldots$ be a sequence constructed as follows: $\mathcal{V}_{\mathcal{T}}^1 = \mathcal{I}_{\star}(\mathcal{T})$ and, for every $i \geq 2$, $\mathcal{V}_{\mathcal{T}}^i$ is obtained by applying some (randomly chosen) rule in Figure 4.1 to $\mathcal{V}_{\mathcal{T}}^{i-1}$. Claims (1-4) can be proven

via induction on the sequence $\mathcal{V}^1_{\mathcal{T}}, \mathcal{V}^2_{\mathcal{T}}, \ldots$ *It is clear that the base case holds since* $\mathcal{V}^1_{\mathcal{T}} = \mathcal{I}_\star(\mathcal{R}(\mathcal{T}))$ *and set* $\mathcal{W}_{\mathcal{T}}$ *contains every fact in* $\{A(\star,\star) \mid A \in N_C(\mathcal{T})\} \cup \{R(\star,\star,\star,\star) \mid R \in N_R(\mathcal{T})\}$. *The induction step can be easily verified by doing a case by case analysis of all the possible facts that may be derived when computing some set* $\mathcal{V}^i_{\mathcal{T}}$ *with* $i \geq 2$.

$\mathsf{Q1}(w, y) : \{\mathsf{physicalEntity}(w, z), \mathsf{physicalEntity}(y, z)\}$

$\mathsf{Q2}(x, z) : \{\mathsf{Complex}(z), \mathsf{Pathway}(x), \mathsf{memberPhysicalEntity}(z, w), \mathsf{memberPhysicalEntity}(z, w),$
$\mathsf{participant}(y, z), \mathsf{pathwayComponent}(x, y)\}$

$\mathsf{Q3}(x, z) : \{\mathsf{SequenceSite}(x), \mathsf{featureLocation}(x, w), \mathsf{featureLocation}(x, y), \mathsf{sequenceIntervalBegin}(w, z),$
$\mathsf{sequenceIntervalBegin}(y, z), \mathsf{SequenceSite}(z)\}$

$\mathsf{Q4}(x, z) : \{\mathsf{Pathway}(x), \mathsf{Protein}(x), \mathsf{participant}(w, z), \mathsf{participant}(y, z), \mathsf{pathwayComponent}(x, w),$
$\mathsf{pathwayComponent}(x, y)\}$

$\mathsf{Q1}(x, y) : \{\mathsf{cellularComponent}(x, z), \mathsf{cellularComponent}(y, z)\}$

$\mathsf{Q2}(x) : \{\mathsf{translatedFrom}(w, x), \mathsf{NucleotideResource}(x), \mathsf{locatedOn}(x, y), \mathsf{database}(x, z), \mathsf{Database}(z)\}$

$\mathsf{Q3}(x) : \{\mathsf{translatedFrom}(w, y), \mathsf{Resource}(y), \mathsf{translatedFrom}(w, x), \mathsf{Resource}(x), \mathsf{database}(y, z),$
$\mathsf{Database}(z), \mathsf{database}(x, z)\}$

$\mathsf{Q4}(x) : \{\mathsf{CellularComponent}(z), \mathsf{locatedIn}(x, w), \mathsf{cellularComponent}(w, z), \mathsf{locatedIn}(x, y),$
$\mathsf{cellularComponent}(y, z)\}$

$\mathsf{Q1}(x, z) : \{\mathsf{worksFor}(x, y), \mathsf{worksFor}(z, y), \mathsf{publicationAuthor}(x, z)$

$\mathsf{Q2}(x) : \{\mathsf{advisor}(x, y), \mathsf{Faculty}(y), \mathsf{teacherOf}(y, z), \mathsf{Course}(z), \mathsf{memberOf}(y, w), \mathsf{Department}(w)\}$

$\mathsf{Q3}(x, y, z) : \{\mathsf{teacherOf}(y, z), \mathsf{Course}(z), \mathsf{advisor}(x, y), \mathsf{Faculty}(y), \mathsf{takesCourse}(x, z), \mathsf{Student}(x)\}$

$\mathsf{Q4}(x) : \{\mathsf{publicationAuthor}(x, z), \mathsf{publicationAuthor}(x, y), \mathsf{advisor}(z, y), \mathsf{Professor}(y),$
$\mathsf{memberOf}(z, w), \mathsf{memberOf}(y, w)\}$

$\mathsf{Q1}(x, y) : \{\mathsf{takesCourse}(x, z), \mathsf{takesCourse}(y, z)\}$

$\mathsf{Q2}(x) : \{\mathsf{teachingAssistantOf}(x, y), \mathsf{TeachingAssistant}(x), \mathsf{publicationAuthor}(z, x),$
$\mathsf{Book}(z), \mathsf{takesCourse}(w, y), \mathsf{worksFor}(x, v), \mathsf{ResearchGroup}(v)\}$

$\mathsf{Q3}(x, y) : \{\mathsf{isFriendOf}(x, y), \mathsf{UndergraduateStudent}(x), \mathsf{GraduateStudent}(y), \mathsf{like}(x, z)\}$

$\mathsf{Q4}(x, y) : \{\mathsf{hasDoctoralDegreeFrom}(x, z), \mathsf{hasDoctoralDegreeFrom}(y, z), \mathsf{hasMasterDegreeFrom}(x, w),$
$\mathsf{hasMasterDegreeFrom}(y, w), \mathsf{worksFor}(x, v), \mathsf{worksFor}(y, v)\}$

Figure 4.2: Queries for Reactome, Uniprot, LUBM and UOBM.

$\forall$-rule	if	there is some TGD $\rho \in \Psi(\mathcal{T})$
	then	$\mathcal{W}_\mathcal{T} \to \rho_O(\mathcal{W}_\mathcal{T}) \cup \mathcal{W}_\mathcal{T}$
Eq_1-rule	if	there is some $A(t, t') \in \mathcal{W}_\mathcal{T}$ and some s with $\mathsf{Eq}(t, s) \in \mathcal{W}_\mathcal{T}$ and $dep(t) \leq dep(s)$
	then	$\mathcal{W}_\mathcal{T} \to \{A(s, t')\} \cup \mathcal{W}_\mathcal{T}$
Eq_2-rule	if	there is some $R(t, t', u, u') \in \mathcal{W}_\mathcal{T}$ and some s with $\mathsf{Eq}(t, s) \in \mathcal{W}_\mathcal{T}$ and $dep(t) \leq dep(s)$ (resp., $\mathsf{Eq}(u, s) \in \mathcal{W}_\mathcal{T}$ and $dep(u) \leq dep(s)$)
	then	$\mathcal{W}_\mathcal{T} \to \{R(s, t', u, u')\} \cup \mathcal{W}_\mathcal{T}$ (resp., $\{R(t, t', s, u')\} \cup \mathcal{V}_\mathcal{T}$)
Eq_3-rule	if	there are some t, u and s with $\mathsf{Eq}(u, t) \in \mathcal{W}_\mathcal{T}$, $\{\mathsf{Eq}(t, s), \mathsf{Eq}(s, u)\} \subseteq \mathcal{W}_\mathcal{T}$ or $t = u$
	then	$\mathcal{W}_\mathcal{T} \to \{\mathsf{Eq}(t, u)\} \cup \mathcal{W}_\mathcal{T}$

Figure 4.3: Expansion rules for the construction of $\mathcal{W}_\mathcal{T}$.

5

The RSA Fragment of Horn DL

In recent years there has been a growing interest in so-called lightweight DL languages, which are based on logics with favorable computational properties. The most prominent examples of lightweight ontology languages are the $\mathcal{EL}^{++}$, DL-Lite$_R$ and DLP (formally introduced in Section 3.2). These languages are often referred to as *profile languages*, because they serve as the formal basis for the profile fragments EL, QL and RL of the Web Ontology Language (OWL) [Hitzler et al. 2009]. Standard reasoning tasks, such as classification and fact entailment, are feasible in polynomial time for all profiles, and many highly scalable profile-specific reasoners have been developed [Baader et al. 2006; Bishop et al. 2011; Calvanese et al. 2011; Kazakov et al. 2014; Motik et al. 2014; Rodriguez-Muro and Calvanese 2012]. In contrast to the lightweight logics underpinning the profiles, the logics required to capture expressive Horn ontologies are intractable: standard reasoning is ExpTime-complete for Horn-$\mathcal{SHOIQ}$ and and 2-ExpTime-completefor the more expressive Horn-$\mathcal{SROIQ}$ [Kazakov 2008].

In an attempt to push the expressivity of lightweight DLs whilst still preserving its beneficious complexity properties, we present the *role safety acyclic* (*RSA*) fragment of Horn DL. This fragment, for which standard reasoning tasks can be solved in polynomial time, encompasses the profile languages in terms of expressiveness; i.e., every ontology defined over a profile language is also an RSA ontology. As in the previous chapter, we also present a reasoning algorithm to solve standard reasoning tasks over this fragment and moreover, we empirically verify that many real-world ontologies, which cannot be defined within the profile languages, are RSA.

5.1 The Notion of Role Safety

As mentioned in the previous chapter, reasoning over Horn-$\mathcal{SROIQ}$, the most expressive of the Horn DL fragments, is intractable. In particular, satisfiability is ExpTime-hard already for Horn-$\mathcal{ALCI}$ (the fragment of Horn-$\mathcal{SROIQ}$ without nominals, cardinality restrictions or complex roles).

A closer look at existing complexity results reveals that the main source of intractability is the phenomenon typically known as *and-branching*: due to the interaction between existential quantifiers over a role R (i.e., axioms of the form $A \sqsubseteq \exists R.B$) and universal quantifiers over R (encoded by axioms of type $A \sqsubseteq \forall R.B$), an ontology may only be satisfied in models of exponential size which cannot be summarized. The same effect can be achieved via the interaction between existential quantifiers and

$$\mathsf{Lazy} \sqsubseteq \mathsf{Student} \tag{5.1}$$

$$\mathsf{Student} \sqsubseteq \exists \mathsf{Attends.Course} \tag{5.2}$$

$$\exists \mathsf{Attends.MorningCourse} \sqsubseteq \mathsf{Diligent} \tag{5.3}$$

$$\mathsf{Lazy} \sqcap \mathsf{Diligent} \sqsubseteq \bot \tag{5.4}$$

$$\mathsf{Course} \sqsubseteq \exists \mathsf{IsAttendedBy.Student} \tag{5.5}$$

$$\mathsf{Attends}^- \sqsubseteq \mathsf{IsAttendedBy} \tag{5.6}$$

$$\mathsf{IsAttendedBy}^- \sqsubseteq \mathsf{Attends} \tag{5.7}$$

$$\mathsf{Lazy}(\mathsf{David}) \tag{5.8}$$

Figure 5.1: Example ontology $\mathcal{O}$

cardinality restrictions (axioms of type $(A \sqsubseteq \leq 1\, R.B)$): reasoning in the extension of the EL profile with counting is also known to be ExpTime-hard [Baader et al. 2005a].

And-branching can be tamed by precluding the harmful interactions between existential and universal quantifiers, on the one hand and existential quantifiers and cardinality restrictions, on the other hand. If we disallow existential quantifiers altogether (i.e., axioms of the form $A \sqsubseteq \exists R.B)$), then we obtain the DLP profile, and ontologies become a subset in terms of expressivity to Datalog programs with equality. Similarly, if we disallow the use of inverse roles and cardinality restrictions, thus precluding both universal quantification over roles and counting, then we obtain the EL profile.

The main idea behind our notion of role safety is to identify a subset of the roles in an ontology over which these potentially harmful interactions between language constructs do not occur. On the one hand, if a role does not occur existentially quantified in axioms of type $A \sqsubseteq R.B$, then its "behavior" is similar to that of a role in an DLP ontology, and hence it is *safe*. On the other hand, if a role occurs existentially quantified, but no axioms involving inverse roles or counting apply to any of its super-roles, then the role behaves like a role in an EL ontology, and hence it is also *safe*.

Definition 5.1.1 *A role R is* safe with respect to some Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ *if at least one of the following conditions holds.*

1. *There are no axioms of the form $A \sqsubseteq \exists R.B \in \mathcal{O}$.*
2. *For every axiom of the form $A \sqsubseteq \leq 1\, S.B \in \mathcal{O}$, $R \not\sqsubseteq^*_{\mathcal{O}} S$ and $R^- \not\sqsubseteq^*_{\mathcal{O}} S$; and for every axiom of the form $\exists S.A \sqsubseteq B \in \mathcal{O}$ with $A \neq \top$, $R^- \not\sqsubseteq^*_{\mathcal{O}} S$.*

See Section 3.2 for the definition of the relation $\sqsubseteq^*_{\mathcal{O}}$. Also, if the ontology $\mathcal{O}$ is clear from the context, we simply say that a role is "safe" instead of "safe with respect to $\mathcal{O}$."

Example 5.1.2 *Consider the example ontology from Figure 5.1, which is not captured by any of the logics underpinning the OWL profile languages. Note that the role* Attends *is safe: although it occurs existentially quantified in axiom (5.2), its inverse* IsAttendedBy *does not occur in an axiom of the form*

$\exists R.A \sqsubseteq B$, and the ontology does not contain at most restrictions. In contrast, the role IsAttendedBy *is unsafe since it occurs existentially quantified in (5.5) and its inverse role* Attends *in (5.3).*

Definition 5.1.1 intuitively explains why Horn-$\mathcal{SHOIQ}$ ontologies captured by any of the logics underpinning the OWL profile languages contain only safe roles:

- In $\mathcal{EL}^{++}$, roles can be existentially quantified, but there are no inverse roles or cardinality restrictions. Thus, the second condition in Definition 5.1.1 trivially holds for every role.
- In DLP, roles do not occur axioms of the form $A \sqsubseteq \exists R.B$. Thus, the first condition of Definition 5.1.1 trivially holds.
- In DL-Lite$_R$, there are no cardinality restrictions and nor axioms of the form $\exists R.A \sqsubseteq B$. Thus, the second condition in Definition 5.1.1 trivially holds.

5.2 Role Safety Acyclicity

In this section, we propose a novel *role safety acyclicity* (RSA) condition that is applicable to Horn-$\mathcal{SHOIQ}$ ontologies and that does not completely preclude unsafe roles. Instead, our condition restricts the way in which unsafe roles are used so that they cannot lead to the interactions between language constructs that are at the root of ExpTime-hardness proofs; in particular, *and-branching*.

To check whether an ontology $\mathcal{O}$ is RSA we first generate a directed graph $G(\mathcal{O})$ by means of a Datalog program $\mathcal{P}_G(\mathcal{O})$. The edges in $G(\mathcal{O})$ are generated from the extension of a fresh "edge" predicate E in the materialization of $\mathcal{P}_G(\mathcal{O})$. Intuitively, the relevant facts over E in the materialization stem from the presence in $\mathcal{O}$ of existential restrictions over unsafe roles. Once the directed graph $G(\mathcal{O})$ has been generated, we check that it is a directed acyclic graph (DAG) and that it does not not contain "diamond-shaped" subgraphs; the former requirement will ensure termination of our reasoning algorithm in Section 5.3, while the latter is critical to ensure tractability. Furthermore, we define a weaker version of RSA (WRSA) where $G(\mathcal{O})$ is only required to be a DAG. Although this relaxed notion does not ensure tractability of reasoning, it does guarantee termination of our reasoning algorithm, and hence is still of relevance in practice.

Definition 5.2.1 *Let $\mathcal{O}$ be an ontology and let π be the mapping defined in Figure 3.2. Let* PE *and* E *be fresh binary predicates, and let* U *be a fresh unary predicate. Furthermore, for each pair of concepts A, B and each role R, let $v^A_{R,B}$ be a fresh constant. Finally, let Ξ be the function mapping each axiom α in $\mathcal{O}$ to a datalog rule as given next, and let $\Xi(\mathcal{O}) = \{\Xi(\alpha) \mid \alpha \text{ in } \mathcal{O}\}$:*

$$\Xi(\alpha) = \begin{cases} A(x) \rightarrow R(x, v^A_{R,B}) \wedge B(v^A_{R,B}) \wedge \mathsf{PE}(x, v^A_{R,B}) & \text{if } \alpha = A \sqsubseteq \exists R.B \\ \pi(\alpha) & \text{Otherwise.} \end{cases}$$

Then, let $\mathcal{P}_G(\mathcal{O})$ be the standard equality axiomatization of the following program

$$\Xi(\mathcal{O}) \cup \{\mathsf{U}(x) \wedge \mathsf{PE}(x, y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x, y)\} \cup \{\mathsf{U}(v^A_{R,B}) \mid R \text{ is unsafe}\}$$

$$\begin{aligned}
\mathsf{Lazy}(x) &\rightarrow \mathsf{Student}(x)\\
\mathsf{Student}(x) &\rightarrow \mathsf{Attends}(x, v^{St}_{At,Co}) \wedge \mathsf{Course}(v^{St}_{At,Co}) \wedge \mathsf{PE}(x, v^{St}_{At,Co})\\
\mathsf{Attends}(x,y) \wedge \mathsf{MorningCourse}(y) &\rightarrow \mathsf{Diligent}(y)\\
\mathsf{Lazy}(x) \wedge \mathsf{Diligent}(x) &\rightarrow \bot(x)\\
\mathsf{Course}(x) &\rightarrow \mathsf{AttendedBy}(x, v^{Co}_{Ia,St}) \wedge \mathsf{Student}(v^{Co}_{Ia,St}) \wedge \mathsf{PE}(x, v^{Co}_{Ia,St})\\
\mathsf{Attends}(y,x) &\rightarrow \mathsf{AttendedBy}(x,y)\\
\mathsf{AttendedBy}(x,y) &\rightarrow \mathsf{Attends}(y,x)\\
\mathsf{U}(x) \wedge \mathsf{PE}(x,y) \wedge \mathsf{U}(y) &\rightarrow \mathsf{E}(x,y)\\
&\mathsf{Lazy}(\mathsf{David})\\
&\mathsf{U}(v^{Co}_{Ia,St})
\end{aligned}$$

Figure 5.2: Checking acyclicity of our example ontology $\mathcal{O}$.

Let $G(\mathcal{O})$ be the smallest directed graph having an edge (c,d) for each fact $\mathsf{E}(c,d)$ with $\mathsf{E}(c,d) \in OC(\mathcal{P}_G(\mathcal{O}))$. Then, we say that $\mathcal{O}$ is Role Safety Acyclic *(RSA) if $G(\mathcal{O})$ is an oriented forest. Furthermore, we say that $\mathcal{O}$ is* weakly RSA *(WRSA) if $G(\mathcal{O})$ is a DAG.*

An oriented forest is a disjoint union of oriented trees; that is, a DAG whose underlying undirected graph is a forest.

The core of the program $\mathcal{P}_G(\mathcal{O})$ is obtained from $\mathcal{O}$ by translating its axioms into first-order logic in the usual way with the single exception of existentially quantified axioms α, which are translated into Datalog by transforming the (unique) existential variable in $\pi(\alpha)$ into a constant. The fresh predicate PE is used to track all facts over roles R generated by the application of the rules, regardless of whether the relevant role R is safe or not. In this way, PE records "possible edges" in the graph. The safety distinction is realised by the unary predicate U, which is populated with all fresh constants introduced by the Skolemisation of existential restrictions over the unsafe roles. Finally, the rule $\mathsf{U}(x) \wedge \mathsf{PE}(x,y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x,y)$ ensures that only possible edges between Skolem constants in the extension of U eventually become edges in the graph.

Example 5.2.2 *Figure 5.2 depicts the rules in the program $\mathcal{P}_G(\mathcal{O})$ for our example ontology $\mathcal{O}$. Consider the application of the chase on $\mathcal{P}_G(\mathcal{O})$, which applies to the initial facts $\mathcal{P}_G(\mathcal{O})^0_O = \{\mathsf{Lazy}(\mathsf{David}), \mathsf{U}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}})\}$. The chase terminates after the following iterations:*

$$\begin{aligned}
\mathcal{P}_G(\mathcal{O})^1_O &= \mathcal{P}_G(\mathcal{O})^0_O \cup \{\mathsf{Student}(\mathsf{David})\}\\
\mathcal{P}_G(\mathcal{O})^2_O &= \mathcal{P}_G(\mathcal{O})^1_O \cup \{\mathsf{Attends}(\mathsf{David}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{Course}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{PE}(\mathsf{David}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}})\}\\
\mathcal{P}_G(\mathcal{O})^3_O &= \mathcal{P}_G(\mathcal{O})^2_O \cup \{\mathsf{AttendedBy}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}), \mathsf{Student}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}), \mathsf{PE}(\mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}, \mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}})\}\\
\mathcal{P}_G(\mathcal{O})^4_O &= \mathcal{P}_G(\mathcal{O})^3_O \cup \{\mathsf{Attends}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}}), \mathsf{PE}(\mathsf{v}^{\mathsf{Co}}_{\mathsf{Ia,St}}, \mathsf{v}^{\mathsf{St}}_{\mathsf{At,Co}})\}
\end{aligned}$$

No more atoms are derived in subsequent steps and hence $OC(P_{\mathcal{O}}) = \mathcal{P}_G(\mathcal{O})^4_O$. Note that the graph

induced by the auxiliary PE *predicate is cyclic; in contrast, the extension of* E *is empty and* $G(\mathcal{O})$ *has no edges. Clearly,* $\mathcal{O}$ *is thus RSA.*

The following theorem establishes that checking RSA and WRSA is tractable. Intuitively, the program $\mathcal{P}(\mathcal{O})$ is linear in the size of $\mathcal{O}$ and each of its rules contains at most three variables regardless of $\mathcal{O}$; as a result, the materialization (and hence also the resulting graph) is polynomially bounded.

Theorem 5.2.3 *Checking whether an ontology* $\mathcal{O}$ *is RSA (resp. WRSA) is feasible in polynomial time in the size of* $\mathcal{O}$*.*

Proof 20 *The program* $\mathcal{P}(\mathcal{O})$ *is linear in the size of* $\mathcal{O}$*. Furthermore, each rule in* $\mathcal{P}(\mathcal{O})$ *contains at most three variables. Thus, the set* $OC(\mathcal{P}(\mathcal{O}))$ *is bounded in size by* $O(|\mathcal{O}|^3)$*. Finally, checking whether a directed graph is an oriented tree (resp. acyclic) is feasible in polynomial time by means of standard graph traversal algorithms.*

5.3 Reasoning Over Acyclic Ontologies

In this section, we show that standard reasoning tasks are tractable for RSA ontologies. To this purpose, we propose a translation from a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ into a set $\mathcal{N}(\mathcal{O})$ of existential rules, which may contain existentially quantified variables in the head. Axioms in $\mathcal{O}$ are translated directly into existential rules as specified in Figure 3.2.

Definition 5.3.1 *Let* $\mathcal{O}$ *be an ontology and let* π *be the mapping defined in Figure 3.2. Furthermore, for each pair of concepts* A, B *and each safe role* R *in* $\mathcal{O}$*, let* $v^A_{R,B}$ *be a fresh constant. Let* Λ *be the function mapping each axiom* α *in* $\mathcal{O}$ *to a Datalog rule as given next:*

$$\Lambda(\alpha) = \begin{cases} A(x) \rightarrow R(x, v^A_{R,B}) \wedge B(v^A_{R,B}) & \text{if } \alpha = A \sqsubseteq \exists R.B \text{ with } R \text{ safe} \\ \pi(\alpha) & \text{Otherwise.} \end{cases}$$

Finally, we define the program $\mathcal{N}(\mathcal{O})$ *as the set* $\{\Lambda(\alpha) \mid \alpha \in \mathcal{O}\}$*.*

Example 5.3.2 *Figure 5.3 depicts the rules of the program* $\mathcal{N}(\mathcal{O})$ *for our running example* $\mathcal{O}$*.*

We next show that this translation preserves satisfiability, subsumption, and instance retrieval reasoning outcomes, regardless of whether the ontology $\mathcal{O}$ is acyclic or not. Thus, we can reason over $\mathcal{N}(\mathcal{O})$ instead of $\mathcal{O}$ without sacrificing correctness.

Lemma 5.3.3 *The following properties hold for each ontology* $\mathcal{O}$*, concept names* A, B *and named individuals* a *and* b*, and* c *is a fresh constant not occurring* $\mathcal{O}$*:*

1. $\mathcal{O}$ *is satisfiable if and only if* $\mathcal{N}(\mathcal{O})$ *is satisfiable if and only if* $OC(\mathcal{N}(\mathcal{O}))$ *contains no fact over* $\bot$*.*

2. $\mathcal{O} \models A(a)$ *if and only if* $\mathcal{N}(\mathcal{O}) \models A(a)$ *if and only if* $A(a) \in OC(\mathcal{N}(\mathcal{O}))$*.*

$$\begin{aligned}
\mathsf{Lazy}(x) &\rightarrow \mathsf{Student}(x)\\
\mathsf{Student}(x) &\rightarrow \mathsf{Attends}(x, v^{St}_{At,Co}) \wedge \mathsf{Course}(v^{St}_{At,Co})\\
\mathsf{Attends}(x,y) \wedge \mathsf{MorningCourse}(y) &\rightarrow \mathsf{Diligent}(y)\\
\mathsf{Lazy}(x) \wedge \mathsf{Diligent}(x) &\rightarrow \bot(x)\\
\mathsf{Course}(x) &\rightarrow \exists y(\mathsf{AttendendBy}(x,y) \wedge \mathsf{Student}(y))\\
\mathsf{Attends}(y,x) &\rightarrow \mathsf{AttendedBy}(x,y)\\
\mathsf{AttendedBy}(x,y) &\rightarrow \mathsf{Attends}(y,x)\\
&\mathsf{Lazy}(\mathsf{David})
\end{aligned}$$

Figure 5.3: Running Example: Reasoning.

3. $\mathcal{O} \models A \sqsubseteq B$ *if and only if* $\mathcal{N}(\mathcal{O}) \cup \{A(c)\} \models B(c)$ *if and only if* $B(c) \in OC(\mathcal{N}(\mathcal{O}) \cup \{A(c)\})$.

So far, we have established that we can dispense with the input ontology $\mathcal{O}$ and reason over the program $\mathcal{N}(\mathcal{O})$ instead. The chase of $\mathcal{N}(\mathcal{O})$, however, may still be infinite. We next show, if $\mathcal{O}$ is RSA, then there exist a polynomial upper bound on the size of the chase of $\mathcal{N}(\mathcal{O})$. Intuitively, every functional term occurring in an atom of the chase of $\mathcal{N}(\mathcal{O})$ corresponds to a single path in $G(\mathcal{O})$, and the size of the graph is polynomial in $\mathcal{O}$. In an oriented forest there is at most one path between any two nodes, which bounds polynomially the number of possible functional terms. In contrast, the latter condition does not hold for DAGs, where only a bound in the length of paths can be guaranteed.

Lemma 5.3.4 *Let* $\mathcal{O}$ *be an RSA ontology. Then, the chase of* $\mathcal{N}(\mathcal{O})$ *terminates and* $OC(\mathcal{N}(\mathcal{O}))$ *is of polynomial size. Furthermore, if* $\mathcal{O}$ *is WRSA, then the chase of* $\mathcal{N}(\mathcal{O})$ *terminates and* $OC(\mathcal{N}(\mathcal{O}))$ *is of exponential size.*

Lemmas 5.3.3 and 5.3.4 suggest a reasoning algorithm for acyclic ontologies $\mathcal{O}$. First, compute the program $\mathcal{N}(\mathcal{O})$ as in Definition 5.3.1. Then, run the chase for $\mathcal{N}(\mathcal{O})$ and read out the reasoning outcomes from the computed Herbrand model. If $G(\mathcal{O})$ is an oriented forest (i.e., $\mathcal{O}$ is RSA) we can implement our algorithm efficiently, which yields the following result as a corollary of the previous theorems.

Theorem 5.3.5 *Satisfiability and unary fact entailment is feasible in polynomial time for the class of RSA ontologies.*

In contrast to RSA, our algorithm runs in exponential time for WRSA ontologies. We next show that, indeed, reasoning with WRSA ontologies is intractable under standard complexity-theoretic assumptions.

Theorem 5.3.6 *Unary fact entailment is* PSPACE*-hard for WRSA ontologies.*

Proof 21 *In [Cuenca Grau et al. 2013] [Lemma 63] validity checking for QBF formulas which is a standard* PSPACE*-complete problem is reduced to fact entailment w.r.t. weakly-acyclic Horn-*$\mathcal{SHI}$

ontologies. While weak-acyclicity and WRSA are two distinct conditions, the particular reduction provided as proof of that lemma produces a set of existential rules which can be translated into a WSRA Horn-$\mathcal{SHI}$ ontology. As such, the reduction shows as well that fact entailment w.r.t. WSRA Horn-$\mathcal{SHOIQ}$ ontologies is PSPACE*-hard.*

5.4 Stronger Notions of Acyclicity

Note that Theorem 5.3.5 does not make any claims about the tractability of concept subsumption for RSA ontologies. To check whether $\mathcal{O} \models A \sqsubseteq B$ we need to extend $\mathcal{N}(\mathcal{O})$ with an assertion $A(c)$ over a fresh individual c, run the Skolem chase, and check whether $B(c)$ is derived (see Lemma 5.3.3). However, as illustrated by the following example, RSA membership is not robust under addition of ABox assertions.

Example 5.4.1 *Let $\mathcal{O}$ consist of a fact $B(c)$ and the following axioms:*

$$A \sqsubseteq B \quad B \sqsubseteq C \quad A \sqsubseteq \exists R.A \quad \top \sqsubseteq \leq 1.R.\top$$

Ontology $\mathcal{O}$ is RSA because the rule corresponding to the "dangerous" axiom $A \sqsubseteq \exists R.A$ involving the unsafe role R does not fire during materialization; as a result, the graph generated by $\mathcal{P}_G(\mathcal{O})$ is empty. Indeed, the chase terminates on $\mathcal{N}(\mathcal{O})$ and determines satisfiability as well as all the facts entailed by $\mathcal{O}$. In contrast, if we add the fact $A(c)$ to $\mathcal{N}(\mathcal{O})$ to determine the subsumers of A, the chase will no longer terminate because the ontology $\mathcal{O}$ extended with $A(c)$ is now cyclic.

To ensure tractability of subsumption and classification, we therefore propose the following stronger notion of acyclicity.

Definition 5.4.2 *Let $\mathcal{O}$ be some ontology. For each concept name A in $\mathcal{O}$, let c_A be a fresh constant and let $\mathcal{A}_{\mathsf{Cl}} = \{A(c_A) \mid A$ is a concept name occurring in $\mathcal{O}\}$. We say that $\mathcal{O}$ is RSA for classification if $\mathcal{O}$ extended with $\mathcal{A}_{\mathsf{Cl}}$ is RSA.*

Tractability of subsumption immediately follows from our results in Section 5.3.

Proposition 5.4.3 *Checking whether $\mathcal{O} \models A \sqsubseteq B$ is feasible in polynomial time for ontologies $\mathcal{O}$ that are RSA for classification.*

Although this notion is well-suited for TBox reasoning, data-intensive applications where the ABox changes frequently require a further strengthening.

Definition 5.4.4 *An ontology $\mathcal{O}$ is* universally RSA *if $\mathcal{O} \cup \mathcal{A}'$ is RSA for every ABox $\mathcal{A}'$.*

Checking whether $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$ is universally RSA can be reduced to checking whether the ontology $\mathcal{O}$ extended with the *critical instance* $\mathcal{I}_\star(\mathcal{A})$ which is defined as the set of facts containing all facts of the form $A(\star)$ and $R(\star, \star)$ for every concept name A and role R occurring in $\mathcal{A}$.

Proposition 5.4.5 *An ontology $\mathcal{O}$ is* universally RSA *if and only if $\mathcal{O} \cup \mathcal{I}_\star(\mathcal{A})$ is RSA.*

Proof 22 *Assume $\mathcal{O}$ is* universally RSA. *Then, it is RSA also for $\mathcal{O} \cup \mathcal{I}_\star(\mathcal{A})$. It remains to be shown that if $\mathcal{O} \cup \mathcal{I}_\star(\mathcal{A})$ is RSA, $\mathcal{O} \cup \mathcal{A}$ is RSA for every ABox $\mathcal{A}$. Let $\mathcal{O}^{\mathcal{A}}$ be the extension of $\mathcal{O}$ with an arbitrary such ABox $\mathcal{A}$ and let $\mathcal{O}^\star = \mathcal{O} \cup \mathcal{I}_\star(\mathcal{O})$. Also let $N_I^\star$ and $N_I^{\mathcal{A}}$ be the set of named individuals occurring in $\mathcal{O}^\star$ and $\mathcal{O}^{\mathcal{A}}$, respectively. Then, we define a mapping $\mu : terms(\mathcal{P}_G(\mathcal{O}^{\mathcal{A}})) \to terms(\mathcal{P}_G(\mathcal{O}^\star))$ as follows:*

$$\mu(x) = \begin{cases} x, & \text{if } x \in terms(\mathcal{P}_G(\mathcal{O}^\star)) \\ \star, & otherwise \end{cases}$$

It can be shown by induction on the level of atoms in $OC(\mathcal{P}_G(\mathcal{O}^{\mathcal{A}}))$ that:

- *for every $A(t) \in OC(\mathcal{P}_G(\mathcal{O}^{\mathcal{A}}))$: $A(\mu(t)) \in OC(\mathcal{P}_G(\mathcal{O}^\star))$, and*
- *for every $R(t,u) \in OC(\mathcal{P}_G(\mathcal{O}^{\mathcal{A}}))$: $R(\mu(t),\mu(u)) \in OC(\mathcal{P}_G(\mathcal{O}^\star))$.*
- *for every $x \approx y \in OC(\mathcal{P}_G(\mathcal{O}^{\mathcal{A}}))$: $\mu(x) \approx \mu(y) \in OC(\mathcal{P}_G(\mathcal{O}^\star))$.*

Thus, the graph $G(\mathcal{O}^{\mathcal{A}})$ is a subgraph of $G(\mathcal{O}^\star)$.

Example 5.4.6 *The critical ABox for our example ontology $\mathcal{O}$ consists of all facts $A(\star)$ and $R(\star,\star)$ for A a concept name and R a role name from $\mathcal{O}$. It can be checked that $\mathcal{O}$ is universally RSA, and hence also RSA for classification.*

Universal RSA is, however, a rather strict condition, especially in the presence of equality. The following example illustrates that, e.g., every ontology with a functional role used in an existential restriction is not universally RSA.

Example 5.4.7 *Consider $\mathcal{O}$ consisting of axioms $A \sqsubseteq \exists R.B$ and $\top \sqsubseteq \leq 1R.\top$. The critical ABox contains facts $A(\star)$, $B(\star)$, and $R(\star,\star)$. The corresponding Datalog program entails a fact $R(\star, v^A_{R,B})$ due to axiom $A \sqsubseteq \exists R.B$. Due to the functionality of R, the individuals $*$ and $v^A_{R,B}$ become equal, and hence we have $A(v^A_{R,B})$ and eventually also $R(v^A_{R,B}, v^A_{R,B})$. Since R is unsafe, the graph contains a cyclic edge $\mathsf{E}(v^A_{R,B}, v^A_{R,B})$. Indeed, the chase of both $\mathcal{O}$ and $\mathcal{N}(\mathcal{O})$ is infinite.*

It is well-known that the Skolem chase often does not terminate in the presence of equality [Cuenca Grau et al. 2013; Marnette 2009]. The standard approach to circumvent this issue is to exploit the so-called *singularization technique* [Marnette 2009] described in Section 3.4.2. After application of the singularization transformation, the ontology is thus equality-free. singularization preserves reasoning outcomes in a well-understood way, and it is effective in addressing non-termination problems.

5.5 Related Work

In recent years the computational properties of Horn DL have been extensively investigated. The logical underpinnings for the EL and QL profiles of OWL 2 are provided by, respectively, the Horn logics $\mathcal{EL}^{++}$[Baader et al. 2005a] and DL-Lite$_R$ [Calvanese et al. 2007], while the RL profile is based

on Datalog and its intersection with DL [Grosof et al. 2003]. Hustadt et al. proposed the expressive logic Horn-$\mathcal{SRIQ}$, and established its complexity [Hustadt et al. 2005]. Krötzsch et al. studied the complexity of a wide range of Horn DLs with complexities in-between the tractable logics underpinning the profiles and Horn-$\mathcal{SROIQ}$ [Krötzsch et al. 2013; Krötzsch et al. 2007]. Finally, the exact complexity of Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ was determined by Ortiz et al. [Ortiz et al. 2010].

Our techniques in Section 5.3 extend the so-called combined approach to reasoning in EL [Kontchakov et al. 2011; Stefanoni et al. 2013], where ontologies are transformed into Datalog programs by means of Skolemisation of all existentially quantified variables into constants. Skolemisation into constants was also exploited by Zhou et al. [Zhou et al. 2015] to compute upper bounds to query answers.

Finally, in the literature we can find a wide range of acyclicity conditions that are sufficient to ensure chase termination. Weak acyclicity [Fagin et al. 2005] was one of the first such notions, and was subsequently extended to joint acyclicity [Krötzsch and Rudolph 2011], acyclicity of a graph of rule dependencies [Baget et al. 2014], and super-weak acyclicity [Marnette 2009], amongst others. The notion of acyclicity closest to ours is model summarizing acyclicity (MSA) [Cuenca Grau et al. 2013], where acyclicity can also be determined by the materialization of a Datalog program. Unlike existing acyclicity notions, ours was designed to ensure tractability of reasoning rather than chase termination. In particular, the Skolem chase of our example RSA ontology $\mathcal{O}$ is infinite and hence $\mathcal{O}$ cannot be captured by any acyclicity condition designed for chase termination. Instead, our notion ensures termination of the Skolem chase over a particular *transformed* Horn program $\mathcal{N}(\mathcal{O})$, which we can use for reasoning over $\mathcal{O}$. Another important difference is that, in contrast to the chase of $\mathcal{O}$, the chase of the transformed program $\mathcal{N}(\mathcal{O})$ is not a universal model of $\mathcal{O}$, and hence it does not preserve answers to general CQs (but only for satisfiability and fact entailment). Finally, although existing acyclicity conditions guarantee termination of the chase, none of them ensures polynomiality of the computed Herbrand model. Indeed, checking fact entailment over Horn-$\mathcal{SHI}$ ontologies that are weakly acyclic [Fagin et al. 2005] (the most basic acyclicity notion for chase termination) is PSPACE-hard [Cuenca Grau et al. 2013].

5.6 Proof of Concept

We have implemented RSA and WRSA checkers using RDFox [Motik et al. 2014] as a Datalog reasoner. For testing, we used the ontologies in the Oxford Repository and the Design Patterns repository. The former is a large repository currently containing 761 real-world ontologies; the latter contains a wide range of smaller ontologies that capture design patterns commonly used in ontology modeling (these ontologies are particularly interesting as they highlight common interactions between language constructs). Experiments were performed on a laptop with 16 GB RAM and an Intel Core 2.9 GHz processor running Java v.1.7.0_21, with a timeout of 30 min.

Our results are summarised in Table 5.1. For each repository, we first selected those ontologies that are Horn-$\mathcal{SHOIQ}$ and are not captured by any of the OWL 2 profiles. We found 126 such ontologies in the Oxford Repository and 23 in the Design Patterns repository. We then tested our

Repository	Reasoning Task	Total	Safe	RSA		Cyclic		Time-out	
				no Sng.	Sng.	no Sng.	Sng.	no Sng.	Sng.
Oxford Ontology Repository	Satisfiability	126	37	37+43	37+44	46	39	0	6
	Classification	126	37	37+35	37+35	52	49	2	5
	Universality	126	37	37+2	37+31	87	57	0	1
Ontology Design Patterns	Satisfiability	23	14	14+9	14+9	0	0	0	0
	Classification	23	14	14+8	14+8	1	1	0	0
	Universality	23	14	14+4	14+8	5	1	0	0

Table 5.1: Acyclicity evaluation results for ontologies outside the OWL 2 profiles.

acyclicity conditions for satisfiability (Def. 5.2.1), classification (Def. 5.4.2) and universality (Def. 5.4.4) on all these ontologies (for classification and universality, we disregarded the ABox part of the ontologies). We performed tests both with and without employing singularization. Interestingly, in both repositories we could not find any ontology that is WRSA but not RSA, and hence the two notions coincided for all our tests.

As we can observe, 37 ontologies in the Oxford Repository contained only safe roles, and hence are RSA. Without singularization, we found 43 additional ontologies with unsafe roles that are RSA, 35 of which were also RSA for classification and only 2 universally acyclic. When using singularization the number of additional RSA ontologies increased significantly, and we obtained 29 additional universally RSA ontologies, but unfortunately our tests timed-out for several ontologies. This can be explained by the fact that the use of singularization leads to more complicated Datalog rules for which RDFox is not optimised.

In the case of the Design Patterns repository, all ontologies are RSA. We only found one ontology that was not universally RSA when using singularization. Ontologies in this repository are smaller, and we encountered no time-outs.

Note that, in this chapter, we do not include an evaluation of the reasoning algorithm. This is because, being so similar to the procedure presented in the previous section, we foresee this evaluation not being very informative. Thus, to gain an insight of the performance of our algorithm presented in this section, see the results presented in [**?**].

Furthermore, the algorithm presented in this section may only be used to solve standard reasoning tasks over RSA ontologies. To solve the problem of CQA over this fragment see [Feier et al. 2015].

5.7 Proofs

Lemma 5.3.3 *The following properties hold for each ontology $\mathcal{O}$, concept names A, B and named individuals a and b, where c is a fresh constant.*

1. *$\mathcal{O}$ is satisfiable if and only if $\mathcal{N}(\mathcal{O})$ is satisfiable if and only if $OC(\mathcal{N}(\mathcal{O}))$ contains no fact over $\bot$.*

2. $\mathcal{O} \models A(a)$ *if and only if* $\mathcal{N}(\mathcal{O}) \models A(a)$ *if and only if* $A(a) \in OC(\mathcal{N}(\mathcal{O}))$.

3. $\mathcal{O} \models A \sqsubseteq B$ *if and only if* $\mathcal{N}(\mathcal{O}) \cup \{A(c)\} \models B(c)$ *if and only if* $B(c) \in OC(\mathcal{N}(\mathcal{O}) \cup \{A(c)\})$.

Proof 23 *For each claim of the form* A iff B iff C *in the theorem it is enough to show that* A iff C *as* B iff C *follows from the properties of the chase (see Section 3.5). We also reformulate all* 'A' *statements regarding satisfiability of and entailments w.r.t.* $\mathcal{O}$ *in terms of properties of the chase of* $\mathcal{O}$. *For the third claim in the theorem, we note that for a Horn ontology* $\mathcal{O}$, *it is well-known that* $\mathcal{O} \models A \sqsubseteq B$ *iff* $\mathcal{O} \cup \{A(c)\} \models B(c)$, *where* c *is a fresh constant. It remains to be shown that:*

a) $OC(\mathcal{P}(\mathcal{O}))$ *contains no fact over* $\bot$ *iff* $OC(\mathcal{N}(\mathcal{O}))$ *contains no fact over* $\bot$; *and*

b) $A(a) \in OC(\mathcal{P}(\mathcal{O}))$ *iff* $A(a) \in OC(\mathcal{N}(\mathcal{O}))$.

To prove the 'if' part of these claims (soundness) we map each term occurring in $OC(\mathcal{N}(\mathcal{O}))$ *to a set of terms occurring in* $OC(\mathcal{P}(\mathcal{O}))$ *and show inductively that certain properties hold between atoms/terms in* $OC(\mathcal{N}(\mathcal{O}))$ *and atoms over mapped terms/mapped terms in* $OC(\mathcal{P}(\mathcal{O}))$.

We first introduce some notions which will make the formulation of the IH more straightforward.

For a Horn-$\mathcal{SHOIQ}$ *ontology* $\mathcal{O}$, *its skolemization* $sk(\mathcal{O})$ *is the program obtained from* $\mathcal{P}(\mathcal{O})$ *by standard Skolemisation of existentially quantified variables into functional terms. For a Horn program* P, *its grounding,* $ground(P)$, *is the program obtained by replacing each variable occurring in* P *with each term that can be formed using constants and functional symbols occurring in* P. *The derivation level of a ground atom* a *in* $OC(P)$, $level(a, OC(P))$, *is a natural number* k *s.t.:* $a \in \mathcal{P}(\mathcal{O})^k_O$ *and* $a \notin \mathcal{P}(\mathcal{O})^{k-1}_O$, *where* S *is the set of facts occurring in* P *and* $\mathcal{H}$ *is the set of rules occurring in* P. *The derivation level of a ground term* t *in* $OC(P)$, $level(t, OC(P))$, *is a natural number* k *s.t.:* t *occurs in some atom* a *with* $level(a, OC(P)) = k$, *but* t *does not occur in any atom* a *with* $level(a, OC(P)) < k$. *For a set of ground atoms* S, $terms(S)$ *is the set of all terms occurring in some atom in* S.

Definition 5.7.1 *Let* $\mathcal{O}$ *be a Horn-*$\mathcal{SHOIQ}$ *ontology, and let* R *be a role name occurring in* $\mathcal{O}$. *We say that* R *is a* forward-sound *role iff for every axiom of type* $A \sqsubseteq \exists S.B$ *in* $\mathcal{O}$, *with* S *being a safe role:* $S \not\sqsubseteq^*_{\mathcal{O}} R$. *Conversely,* R *is a* backward-sound *role iff for every axiom of type* $A \sqsubseteq \exists S.B$ *in* $\mathcal{O}$, *with* S *being a safe role:* $S \not\sqsubseteq^*_{\mathcal{O}} R^-$.

Lemma 5.7.2 *Let* $\mathcal{O}$ *be a Horn-*$\mathcal{SHOIQ}$ *ontology and let* $\mu : terms(OC(\mathcal{N}(\mathcal{O}))) \to 2^{terms(OC(\mathcal{P}(\mathcal{O})))}$ *be the following function:*

$$\mu(x) = \begin{cases} \{x\}, & \text{if } x \in N_I \\ \{f^C_{RD}(t) \mid t \in \mu(y)\}, & \text{if } x = f^C_{RD}(y) \\ \{f^C_{RD}(y) \mid f^C_{RD}(y) \in terms(OC(\mathcal{P}(\mathcal{O})))\} & \text{if } x = v^C_{RD} \end{cases}$$

Then, all of the the following hold.

i) *For every* $x \in terms(OC(\mathcal{N}(\mathcal{O})))$: $\mu(x) \neq \emptyset$.

ii) $A(x) \in OC(\mathcal{N}(\mathcal{O}))$ *implies* $A(t) \in OC(\mathcal{P}(\mathcal{O}))$, *for every* $t \in \mu(x)$ *and unary predicate* $A \in N_C$.

iii) *$R(x,y) \in OC(\mathcal{N}(\mathcal{O}))$, where R is a backward-sound role implies: for every $t_1 \in \mu(x)$, there exists a $t_2 \in \mu(y)$ s.t. $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$.*

iv) *$R(x,y) \in OC(\mathcal{N}(\mathcal{O}))$, where R is forward-sound role implies: for every $t_2 \in \mu(y)$, there exists a $t_1 \in \mu(x)$ s.t. $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$.*

v) *$R(x,y) \in OC(\mathcal{N}(\mathcal{O}))$, where R is a simple role implies: for every $t_1 \in \mu(x)$, there exists a $t_2 \in \mu(y)$ s.t. $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$, and for every $t_2 \in \mu(y)$, there exists a $t_1 \in \mu(x)$ s.t. $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$.*

vi) *$x \approx y \in OC(\mathcal{N}(\mathcal{O}))$ implies: for every $t_1 \in \mu(x)$, there exists a $t_2 \in \mu(y)$ s.t. $t_1 \approx t_2 \in OC(\mathcal{P}(\mathcal{O}))$, and for every $t_2 \in \mu(y)$, there exists a $t_1 \in \mu(x)$ s.t. $t_1 \approx t_2 \in OC(\mathcal{P}(\mathcal{O}))$.*

Proof 24 *By induction on the derivation level of atoms and terms in $OC(\mathcal{N}(\mathcal{O}))$.*

IB: the hypothesis holds for every ABox assertion, named individual $a \in N_I$ and facts of type $x \approx x \in OC(\mathcal{N}(\mathcal{O}))$.

IH: the hypothesis holds for every atom/term a with $level(a, OC(\mathcal{N}(\mathcal{O}))) < k$. We show that it holds also for every atom/term a with $level(a, OC(\mathcal{N}(\mathcal{O}))) = k$:

i) *$a \in terms(OC(\mathcal{N}(\mathcal{O})))$ (other than some $i \in N_I$). Then a is either of the form:*

1. *v^C_{RD}: then, it has been introduced in $OC(\mathcal{N}(\mathcal{O}))$ via a rule of the form $C(x) \rightarrow R(x, v^C_{RD}) \wedge D(v^C_{RD})$ and $sk(\mathcal{O})$ contains a counterpart rule $C(x) \rightarrow R(x, f^C_{RD}(x)) \wedge D(f^C_{RD}(x))$(†). Then: $level(x, OC(\mathcal{N}(\mathcal{O}))) < level(a, OC(\mathcal{N}(\mathcal{O})))$. From the IH: $\mu(x) \neq \emptyset$ and for every $y \in \mu(x)$: $C(y) \in OC(\mathcal{P}(\mathcal{O}))$. Thus, there exists a u s.t. $C(u) \in OC(\mathcal{P}(\mathcal{O}))$ and from (†) it follows that: $D(f^C_{RD}(u)) \in OC(\mathcal{P}(\mathcal{O}))$. Then $f^C_{RD}(u) \in \mu(v^C_{RD})$, and thus $\mu(v^C_{RD}) \neq \emptyset$.*
2. *or of the form $f^C_{RD}(y)$. From the IH: $\mu(y) \neq \emptyset$ and thus, $\mu(f^C_{RD}(y)) \neq \emptyset$.*

ii) *a is of the form $A(x)$. Then, $\mathcal{N}(\mathcal{O})$ must contain a rule with head a whose body is satisfied in $OC(\mathcal{N}(\mathcal{O}))$:*

1. *$C_1(x) \wedge \ldots \wedge C_n(x) \rightarrow D(x)$ (where $a = D(x)$): from the IH, for every $t \in \mu(x)$: $C_1(t), \ldots, C_n(t) \in OC(\mathcal{P}(\mathcal{O}))$. Then, by applying the counterpart rule in $sk(\mathcal{O})$ we obtain that for every $t \in \mu(x)$: $D(t) \in OC(\mathcal{P}(\mathcal{O}))$.*
2. *$C(x) \rightarrow R(x, f^C_{RD}(x)) \wedge D(f^C_{RD}(x))$ (where $a = D(f^C_{RD}(x))$): from the IH, for every $t \in \mu(x)$: $C(t) \in OC(\mathcal{P}(\mathcal{O}))$. Then, for every $t \in \mu(x)$: $D(f^C_{RD}(t)) \in OC(\mathcal{P}(\mathcal{O}))$, or for every $t \in \mu(f^C_{RD}(x))$: $D(t) \in OC(\mathcal{P}(\mathcal{O}))$.*
3. *$C(x) \rightarrow R(x, v^C_{RD}) \wedge D(v^C_{RD})$ (where $a = D(v^C_{RD})$). Then, there exists a GCI of type: $C \sqsubseteq \exists R.D$ in $\mathcal{O}$ and $sk(\mathcal{O})$ contains a rule of type $C(x) \rightarrow D(f^C_{RD}(x))$. Note that this is the only rule which introduces functional terms of type $f^C_{RD}(\ldots)$. Thus, for every such term $t = f^C_{RD}(y)$ occurring in $terms(OC(\mathcal{P}(\mathcal{O})))$ it holds that $D(t) \in OC(\mathcal{P}(\mathcal{O}))$.But $\mu(v^C_{RD})$ is exactly the set of all such terms.*

4. $R(x,y) \wedge C(y) \rightarrow D(x)$ *(where* $a = D(x)$*). Then,* R *must be a backward-sound role (from the definition of safe roles). From the IH: for every* $t \in \mu(x)$*,* $C(t) \in OC(\mathcal{P}(\mathcal{O}))$ *and there exists a* $t' \in \mu(y)$ *s.t.* $R(t,t') \in OC(\mathcal{P}(\mathcal{O}))$*. Then, by applying the counterpart rule in* $sk(\mathcal{O})$ *for every* $t \in \mu(x)$ *we obtain* $D(t) \in OC(\mathcal{P}(\mathcal{O}))$*.*

5. $C(x) \wedge x \approx y \rightarrow C(y)$*. From the IH: for every* $t \in \mu(x)$*,* $C(t) \in OC(\mathcal{P}(\mathcal{O}))$ *and for every* $t_2 \in \mu(y)$ *there exists a* $t_1 \in \mu(x)$ *s.t.* $t_1 \approx t_2$*. Then,* $C(t_1) \in OC(\mathcal{P}(\mathcal{O}))$ *for every such* t_1*, and by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$ *we obtain* $C(t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, for every* $t_2 \in \mu(y)$*.*

iii) $a = R(x,y)$*, where* R *is a backward-sound role. Then, there must be a ground rule with head* $R(x,y)$ *whose body is satisfied in* $ground(\mathcal{N}(\mathcal{O}))$*:*

1. $U(x,y) \rightarrow R(x,y)$*. As* R *is a backward-sound role,* U *is a backward-sound role as well. From the IH: for every* $t_1 \in \mu(x)$*, there exists* $t_2 \in \mu(y)$ *s.t.* $U(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and thus, by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*: for every* $t_1 \in \mu(x)$*, there exists* $t_2 \in \mu(y)$ *s.t.* $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*.*

2. $U(y,x) \rightarrow R(x,y)$*. Then,* $U^- \sqsubseteq^*_{\mathcal{O}} R$*, and* U *must be a forward-sound role (otherwise* R *would not be a backward-sound role). Then from the IH: for every* $t_2 \in \mu(x)$*, there exists* $t_1 \in \mu(y)$ *s.t.* $U(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and thus by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*: for every* $t_2 \in \mu(x)$*, there exists* $t_1 \in \mu(y)$ *s.t.* $R(t_2,t_1) \in OC(\mathcal{P}(\mathcal{O}))$*.*

3. $C(x) \rightarrow R(x, f^C_{RD}(x)) \wedge D(f^C_{RD}(x))$*. Similar to case ii) 2) above: for every* $t \in \mu(x)$*:* $R(t, f(t)) \in OC(\mathcal{P}(\mathcal{O}))$*.*

4. $C(x) \rightarrow R(x, v^C_{RD}) \wedge D(v^C_{RD})$*. Then, there exists a GCI of type:* $C \sqsubseteq \exists R.D$ *in* $\mathcal{O}$ *and* $sk(\pi(\mathcal{O}))$ *contains a rule of type* $C(x) \rightarrow D(f^C_{RD}(x))$ *(*$\dagger$*). From the IH: for every* $t \in \mu(x)$*:* $C(t) \in OC(\mathcal{P}(\mathcal{O}))$*. Then, by applying (*$\dagger$*) we obtain* $R(t, f^C_{RD}(t)) \in OC(\mathcal{P}(\mathcal{O}))$*, for every* $t \in \mu(x)$*).*

5. $R(x,s) \wedge R(s,y) \rightarrow R(x,y)$*. From the IH it follows that: for every* $t_1 \in \mu(x)$*, there exists* $t_2 \in \mu(s)$ *s.t.* $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$ *and for every* $t_2 \in \mu(s)$*, there exists* $t_3 \in \mu(y)$ *s.t.* $R(t_2,t_3) \in OC(\mathcal{P}(\mathcal{O}))$*. By applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*, we obtain that for every* $z \in \mu(x)$ *there exists* $u \in \mu(y)$ *s.t.* $R(z,u) \in OC(\mathcal{P}(\mathcal{O}))$*.*

6. $R(x,y) \wedge x \approx z \rightarrow R(z,y)$*. From the IH: for every* $t_1 \in \mu(x)$*, there exists* $t_2 \in \mu(y)$ *s.t.* $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and for every* $t_3 \in \mu(z)$*, there exists* $t_1 \in \mu(x)$ *s.t.* $t_1 \approx t_3$*. Then, by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*, we obtain that for every* $t_3 \in \mu(z)$ *there exists* $t_2 \in \mu(y)$ *s.t.* $R(t_3,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*.*

7. $R(x,y) \wedge y \approx z \rightarrow R(x,y)$*. From the IH: for every* $t_1 \in \mu(x)$*, there exists* $t_2 \in \mu(y)$ *s.t.* $R(t_1,t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and for every* $t_2 \in \mu(y)$*, there exists* $t_3 \in \mu(z)$ *s.t.* $t_2 \approx t_3$*. Then, by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*, we obtain that for every* $t_1 \in \mu(x)$ *there exists* $t_3 \in \mu(y)$ *s.t.* $R(t_1,t_3) \in OC(\mathcal{P}(\mathcal{O}))$*.*

iv) $a = R(x, y)$, *where* R *is a forward-sound role. Then, there must be a ground rule with head* $R(x, y)$ *whose body is satisfied in* $ground(\mathcal{N}(\mathcal{O}))$*:*

1. $U(x, y) \rightarrow R(x, y)$. *Similar to case iii) 1) above.*
2. $U(y, x) \rightarrow R(x, y)$. *Then,* $U^- \sqsubseteq^*_{\mathcal{O}} R$ *and thus,* U *is a backward-sound role. From the IH: for every* $t_1 \in \mu(y)$*, there exists* $t_2 \in \mu(x)$ *s.t.* $U(t_1, t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and thus by applying the counterpart rule, for every* $t_1 \in \mu(y)$*, there exists* $t_2 \in \mu(x)$ *s.t.* $R(t_2, t_1) \in OC(\mathcal{P}(\mathcal{O}))$.
3. $C(x) \rightarrow R(x, f^C_{RD}(x)) \wedge D(f^C_{RD}(x))$. *Similar to case iii) 2) above.*
4. $C(x) \rightarrow R(x, v^C_{RD}) \wedge D(v^C_{RD})$*: then* R *must be safe. Contradiction with* R *being a forward-sound role.*
5. $R(x, s) \wedge R(s, y) \rightarrow R(x, y)$. *From the IH it follows that: for every* $t_3 \in \mu(y)$*, there exists* $t_2 \in \mu(s)$ *s.t.* $R(t_2, t_3) \in OC(\mathcal{P}(\mathcal{O}))$ *and for every* $t_2 \in \mu(s)$*, there exists* $t_1 \in \mu(x)$ *s.t.* $R(t_1, t_2) \in OC(\mathcal{P}(\mathcal{O}))$*. By applying the counterpart rule, we obtain that for every* $t_3 \in \mu(y)$ *there exists* $t_1 \in \mu(x)$ *s.t.* $R(t_1, t_3) \in OC(\mathcal{P}(\mathcal{O}))$.
6. $R(x, y) \wedge x \approx z \rightarrow R(z, y)$. *From the IH: for every* $t_2 \in \mu(y)$*, there exists* $t_1 \in \mu(x)$ *s.t.* $R(t_1, t_2) \in OC(\mathcal{P}(\mathcal{O}))$*, and for every* $t_1 \in \mu(x)$*, there exists* $t_3 \in \mu(z)$ *s.t.* $t_1 \approx t_3$*. Then, by applying the counterpart rule in* $sk(\pi(\mathcal{O}))$*, we obtain that for every* $t_2 \in \mu(y)$ *there exists* $t_3 \in \mu(z)$ *s.t.* $R(t_3, t_2) \in OC(\mathcal{P}(\mathcal{O}))$.
7. *Similar to case iii) 7) above.*

v) $a = R(x, y)$*, with* R *being a simple role. Then, there must be a ground rule with head* $R(x, y)$ *whose body is satisfied in* $ground(\mathcal{P}(\mathcal{O}) \uparrow \omega)$*:*

1. $U(x, y) \rightarrow R(x, y)$*:* U *is a simple role as well, follows directly from the IH.*
2. $U(y, x) \rightarrow R(x, y)$*:* U^- *is a simple role as well, follows from the symmetry of the IH.*
3. $C(x) \rightarrow D(v^C_{RD}) \wedge R(x, v^C_{RD})$*: then* R *must be safe and there exists a GCI of type:* $C \sqsubseteq \exists R.D$ *in* $\mathcal{O}$ *and* $sk(\mathcal{O})$ *contains a rule of type* $C(x) \rightarrow D(f^C_{RD}(x)) \wedge R(x, f^C_{RD}(x))$*. Note that this is the only rule which introduces functional terms of type* $f^C_{RD}(\ldots)$*. Thus, for every such term* $t = f^C_{RD}(y)$ *occurring in* $terms(OC(\mathcal{P}(\mathcal{O})))$ *it holds that* $R(y, f^C_{RD}(y)) \in OC(\mathcal{P}(\mathcal{O}))$*.But* $\mu(v^C_{RD})$ *is exactly the set of all such terms. Also, from the IH for every* $t \in \mu(x)$*:* $C(t) \in OC(\mathcal{P}(\mathcal{O}))$*. Then, for every* $t \in \mu(x)$*:* $R(t, f^C_{RD}(t)) \in OC(\mathcal{P}(\mathcal{O}))$.
4. $C(x) \rightarrow D(f^C_{RD}(y) \wedge R(x, f^C_{RD}(y))$*: from the IH, for every* $t \in \mu(x)$*:* $C(t) \in OC(\mathcal{P}(\mathcal{O}))$*. Then, for every* $t \in \mu(x)$*:* $R(t, f^C_{RD}(t)) \in OC(\mathcal{P}(\mathcal{O}))$*, or for every* $t \in \mu(f^C_{RD}(x))$*:* $D(t) \in OC(\mathcal{P}(\mathcal{O}))$.
5. $R(x, y) \wedge x \approx z \rightarrow R(z, y)$. *Similar to cases iii) 6) (in one direction) and iv) 6) (in the other direction) above.*
6. $R(x, y) \wedge y \approx z \rightarrow R(x, z)$. *Similar to cases iii) 7) (in one direction) and iv) 7) (in the other direction) above.*

vi) *a is an equality atom: $a = x \approx y$. Then, there must be a ground rule whose body is satisfied in $ground(\mathcal{P}(\mathcal{O}) \uparrow \omega)$:*

1. *$C(s) \wedge R(s,x) \wedge D(x) \wedge R(s,y) \wedge D(y) \rightarrow x \approx y$: Then, R is a simple role and from the IH:*
   - *for every $t_1 \in \mu(x)$, $D(t_1) \in OC(\mathcal{P}(\mathcal{O}))$ and there exists $t_2 \in \mu(s)$ s.t. $R(t_2, t_1) \in OC(\mathcal{P}(\mathcal{O}))$ and for every $t_2 \in \mu(s)$ there exists $t_3 \in \mu(y)$ s.t. $R(t_2, t_3) \in OC(\mathcal{P}(\mathcal{O}))$. Also, for every $t_2 \in \mu(s)$, $C(t_2) \in OC(\mathcal{P}(\mathcal{O}))$, and for every $t_3 \in \mu(y)$, $D(t_3) \in OC(\mathcal{P}(\mathcal{O}))$. Thus, by applying the counterpart equality rule in $sk(\pi(\mathcal{O}))$, we obtain that for every $t_1 \in \mu(x)$, there exists $t_3 \in \mu(y)$ s.t. $t_1 \approx t_3$;*
   - *similarly as above one can show that for every $t_3 \in \mu(y)$, there exists $t_1 \in \mu(x)$ s.t. $t_1 \approx t_3$;*
2. *$C(x) \rightarrow x \approx a$. From the IH: for every $t \in \mu(x)$, $C(t) \in OC(\mathcal{P}(\mathcal{O}))$ and thus for every $t \in \mu(x)$: $t \approx a \in OC(\mathcal{P}(\mathcal{O}))$. As $\mu(x) \neq \emptyset$ (also from the IH), it follows that for every $t_2 \in \mu(a) = \{a\}$, there exists $t_1 \in \mu(x)$ s.t. $t_1 \approx t_2$.*
3. *$x \approx y \rightarrow y \approx x$: follows from the symmetry of the IH.*
4. *$x \approx y \wedge y \approx z \rightarrow x \approx z$: follows from the IH, similar to case iv) 5), but bidirectional.*

Claims a) and b) follow directly from Lemma 5.7.2 point ii).

Lemma 5.7.3 *Let $\mathcal{O}$ be a Horn-$\mathcal{SHOIQ}$ ontology. If $f^C_{RD}(f^A_{SB}(t)) \in terms(OC(\mathcal{N}(\mathcal{O})))$, then $\mathsf{E}(v^C_{RD}, v^A_{SB}) \in OC(\mathcal{P}_G(\mathcal{O}))$.*

Proof 25 *Let $\mathcal{V} = \{v^C_{RD} \mid v^C_{RD} \in terms(OC(\mathcal{N}(\mathcal{O})))\}$ and μ be the function defined over the set of $terms(OC(\mathcal{N}(\mathcal{O})))$ as follows:*

$$\mu(x) = \begin{cases} x, & \text{if } x \in N_I \cup \mathcal{V} \\ v^C_{RD}, & \text{if } x = f^C_{RD}(y) \end{cases}$$

Then, it can be shown by straightforward induction that: $C(x) \in OC(\mathcal{N}(\mathcal{O}))$ implies $C(x) \in OC(\mathcal{P}_G(\mathcal{O}))$ (all rules in $\mathcal{N}(\mathcal{O})$ are also in $\mathcal{P}_G(\mathcal{O})$ except for rules of type $C(x) \rightarrow R(x, f^C_{RD}(x)) \wedge D(f^C_{RD}(x))$ which are replaced with rules of type $C(x) \rightarrow R(x, v^C_{RD}) \wedge D(v^C_{RD}) \wedge \mathsf{PE}(x, v^C_{RD})$).

Assume $f^C_{RD}(f^A_{SB}(t)) \in terms(OC(\mathcal{N}(\mathcal{O})))$. Then, $ground(\mathcal{N}(\mathcal{O}))$ must contain the following two rules:

- *$A(t) \rightarrow S(t, f^A_{SB}(t)) \wedge B(f^A_{SB}(t))$, and*
- *$C(f^A_{SB}(t)) \rightarrow R(f^A_{SB}(t), f^C_{RD}(f^A_{SB}(t))) \wedge D(f^C_{RD}(f^A_{SB}(t)))$,*

and it must also be the case that: $A(t) \in OC(\mathcal{N}(\mathcal{O}))$ and $C(f^A_{SB}(t)) \in OC(\mathcal{N}(\mathcal{O}))$. Then, $A(\mu(t)) \in OC(\mathcal{P}_G(\mathcal{O}))$, $C(\mu(f^A_{SB}(t))) \in OC(\mathcal{P}_G(\mathcal{O}))$, and $ground(\mathcal{P}_G(\mathcal{O}))$ contains the following rules:

- *$A(\mu(t)) \rightarrow S(\mu(t), v^C_{RD}) \wedge B(v^C_{RD}) \wedge \mathsf{PE}(\mu(t), v^C_{RD})$,*
- *$C(v^C_{RD}) \rightarrow R(v^C_{RD}, v^A_{SB}) \wedge D(v^A_{SB}) \wedge \mathsf{PE}(v^C_{RD}, v^A_{SB})$,*

- $U(v^C_{RD}) \wedge \mathsf{PE}(v^C_{RD}, v^A_{SB}) \wedge U(v^A_{SB}) \rightarrow \mathsf{E}(v^C_{RD}, v^A_{SB})$, *and*
- *facts:* $U(v^C_{RD})$ *and* $U(v^A_{SB})$.

From the above it follows that: $\mathcal{P}_G(\mathcal{O}) \models \mathsf{E}(v^C_{RD}, v^A_{SB})$, *and thus:* $\mathsf{E}(v^C_{RD}, v^A_{SB}) \in OC(\mathcal{P}_G(\mathcal{O}))$.

Lemma 5.3.5 *Let $\mathcal{O}$ be an RSA ontology with signature Σ. Then, the Skolem chase of $\mathcal{N}(\mathcal{O})$ terminates with a Herbrand model of polynomial size. Furthermore, if $\mathcal{O}$ is WRSA, then the Skolem chase of $\mathcal{N}(\mathcal{O})$ terminates with a Herbrand model of size at most exponential.*

Proof 26 *Let $t \in terms(OC(\mathcal{N}(\mathcal{O})))$. Then, t is of the form $g_n(\ldots(g_1(u))\ldots)$, where each g_i is of the form $f^{C_i}_{R_i,D_i}$ and $u \in N_I$ or u is of the form v^C_{RD}. From Lemma 5.7.3 it follows that $\mathsf{E}(v_i, v_{i+1}) \in OC(\mathcal{P}_G(\mathcal{O}))$, where $v_i = f^{C_i}_{R_i,D_i}$, for every $1 \leq i < n$.*

If d is a polytree, for every two nodes v_1 and v_n there is at most one path in $G(\mathcal{O})$: $(v_1, \ldots, v_n)$ which connects them. Thus for given g_n, g_1, and u, $OC(\mathcal{N}(\mathcal{O}))$ contains at most one term t as above. As both the number of function symbols and of terms of form v^C_{RD} in $\mathcal{N}(\mathcal{O})$ is polynomial in the size of the $\mathcal{O}$ and the number of unary and binary atoms which occur in $\mathcal{N}(\mathcal{O})$ is also polynomial, it follows that the size of $OC(\mathcal{N}(\mathcal{O}))$ if also polynomial in the size of $\mathcal{O}$.

If $G(\mathcal{O})$ is acyclic, every path of the form $(v_1, \ldots, v_n)$ in $G(\mathcal{O})$ must not contain the same node twice. Then, the number of terms t of form $g_n(\ldots(g_1(u))\ldots)$ is bounded by ck^n, where c is the number of named individuals and terms of form form v^C_{RD} occurring in $\mathcal{N}(\mathcal{O})$ and k is the number of function symbols occurring in $\mathcal{N}(\mathcal{O})$. Thus, the total number of terms occurring in $\mathcal{N}(\mathcal{O})$ is finite and bounded by $c\Sigma_{0 \leq i \leq k} k^i$, which is exponential in the size of $\mathcal{O}$. Consequently, the size of $OC(\mathcal{N}(\mathcal{O}))$ is also bounded by an exponential in the size of $\mathcal{O}$.

6

Conclusions and Future Work

We have proposed the new classes of RCA_n and RSA ontologies, presented in the previous two chapters. Our experiments suggest that a significant proportion of out-of-profile ontologies are RCA_n and RSA; as a result, we can exploit a efficient algorithms, also presented in the previous sections, to efficiently solve reasoning tasks over ontologies within these fragments.

As for future work, we intend to follow different directions with both approaches presented in the previous chapters.

- First of, we intend to lift RCA_n so it can be applied to the more general fragment of disjunctive existential rules; i.e., rules that, apart from existentially quantified variables in the head, also allow for the use of disjunction. This extension would allow the application of RCA_n to (possibly non-Horn) ontologies, for which there are no implementations that can solve CQ entailment.
- The reasoning techniques presented in the previous chapter may be extended to the whole fragment of Horn-$\mathcal{ALCIOQ}$. Despite the fact that tractability of reasoning would be lost, we foresee that this would enable the development of very efficient reasoning algorithms for this fragment. Furthermore, the notions from [Feier et al. 2015], an implementation to solve CQ entailment over such fragment could also be produced.

7

References

References

Baader, F., Brandt, S., and Lutz, C. 2005a. Pushing the EL envelope. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, L. P. Kaelbling and A. Saffiotti, Eds. Professional Book Center, 364–369.

Baader, F., Brandt, S., and Lutz, C. 2005b. Pushing the EL envelope. In *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*. Morgan-Kaufmann Publishers, Edinburgh, UK.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*, Second ed. Cambridge University Press.

Baader, F., Lutz, C., and Suntisrivaraporn, B. 2006. CEL - A polynomial-time reasoner for life science ontologies. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, U. Furbach and N. Shankar, Eds. Lecture Notes in Computer Science, vol. 4130. Springer, 287–291.

Baget, J., Garreau, F., Mugnier, M., and Rocher, S. 2014. Extending acyclicity notions for existential rules. In *ECAI 2014*. Frontiers in Artificial Intelligence and Applications, vol. 263. IOS Press, 39–44.

Barrett, C. L., Jacob, R., and Marathe, M. V. 2000. Formal-language-constrained path problems. *SIAM J. Comput. 30,* 3, 809–837.

Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., and Velkov, R. 2011. OWLIM: A family of scalable semantic repositories. *Semantic Web 2,* 1, 33–42.

Calì, A., Gottlob, G., and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res. (JAIR) 48*, 115–174.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., and Savo, D. F. 2011. The MASTRO system for ontology-based data access. *Semantic Web 2,* 1, 43–53.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Automated Reasoning 39,* 3, 385–429.

Carral, D. and Hitzler, P. 2012. Extending description logic rules. In *ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*. Lecture Notes in Computer Science, vol. 7295. Springer, 345–359.

Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., and Wang, Z. 2013. Acyclicity notions for existential rules and their application to query answering in ontologies. *JAIR 47*, 741–808.

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. 2005. Data exchange: semantics and query answering. *Theor. Comput. Sci. 336,* 1, 89–124.

Fan, W. 2012. Graph pattern matching revised for social network analysis. In *15th International Conference on Database Theory, ICDT '12, Berlin, Germany, March 26-29, 2012*, A. Deutsch, Ed. ACM, 8–21.

Feier, C., Carral, D., Stefanoni, G., Grau, B. C., and Horrocks, I. 2015. The combined approach to query answering beyond the OWL 2 profiles. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. Wooldridge, Eds. AAAI Press, 2971–2977.

Grosof, B., Horrocks, I., Volz, R., and Decker, S. 2003. Description logic programs: combining logic programs with description logic. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*. ACM, 48–57.

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S., Eds. 27 October 2009. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation. Available at `http://www.w3.org/TR/owl2-primer/`.

Hitzler, P., Krötzsch, M., and Rudolph, S. 2009. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Horrocks, I., Kutz, O., and Sattler, U. 2006. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*. AAAI Press, 57–67.

Hustadt, U., Motik, B., and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 466–471.

Kazakov, Y. 2008. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *11th on Principles of Knowledge Representation and Reasoning (KR'08)*, G. Brewka and J. Lang, Eds. AAAI Press, 274–284.

Kazakov, Y., Krötzsch, M., and Simancik, F. 2014. The incredible ELK - from polynomial procedures to efficient reasoning with el ontologies. *J. Autom. Reasoning 53,* 1, 1–61.

Kontchakov, R., Lutz, C., Toman, D., Wolter, F., and Zakharyaschev, M. 2011. The combined approach to ontology-based data access. See **?**]DBLP:conf/ijcai/2011, 2656–2661.

Krötzsch, M. and Rudolph, S. 2011. Extending decidable existential rules by joining acyclicity and guardedness. In *IJCAI*. 963–968.

Krötzsch, M., Rudolph, S., and Hitzler, P. 2007. Complexity boundaries for Horn description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07)*. AAAI Press, 452–457.

Krötzsch, M., Rudolph, S., and Hitzler, P. 2013. Complexities of Horn description logics. *ACM Trans. Comp. Log. 14,* 1, 2:1–2:36.

Lécué, F., Tallevi-Diotallevi, S., Hayes, J., Tucker, R., Bicer, V., Sbodio, M. L., and Tommasi, P. 2014. Smart traffic analytics in the semantic web with STAR-CITY: scenarios, system and lessons learned in dublin city. *J. Web Sem. 27*, 26–33.

Marnette, B. 2009. Generalized schema-mappings: from termination to tractability. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, J. Paredaens and J. Su, Eds. ACM, 13–22.

Motik, B., Nenov, Y., Piro, R., Horrocks, I., and Olteanu, D. 2014. Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, C. E. Brodley and P. Stone, Eds. AAAI Press, 129–137.

Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., and Banerjee, J. 2015. Rdfox: A highly-scalable RDF store. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*. Lecture Notes in Computer Science, vol. 9367. Springer, 3–20.

Ortiz, M., Rudolph, S., and Simkus, M. 2010. Worst-case optimal reasoning for the horn-dl fragments of OWL 1 and 2. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, F. Lin, U. Sattler, and M. Truszczynski, Eds. AAAI Press.

Ortiz, M., Rudolph, S., and Simkus, M. 2011. Query answering in the horn fragments of the description logics SHOIQ and SROIQ. See **?**]DBLP:conf/ijcai/2011, 1039–1044.

Rodriguez-Muro, M. and Calvanese, D. 2012. High performance query answering over dl-lite ontologies. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*.

Rudolph, S. and Glimm, B. 2014. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *CoRR abs/1401.3849*.

Stefanoni, G., Motik, B., and Horrocks, I. 2013. Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In *AAAI*.

Stefanoni, G., Motik, B., Krötzsch, M., and Rudolph, S. 2014. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *J. Artif. Intell. Res. (JAIR) 51*, 645–705.

Steigmiller, A., Liebig, T., and Glimm, B. 2014. Konclude: System description. *J. Web Sem. 27*, 78–85.

Walsh, T., Ed. 2011. *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. IJCAI/AAAI.

Zhou, Y., Grau, B. C., Nenov, Y., Kaminski, M., and Horrocks, I. 2015. Pagoda: Pay-as-you-go ontology query answering with a datalog reasoner. *J. Artif. Intell. Res. (JAIR) 54*, 309–367.